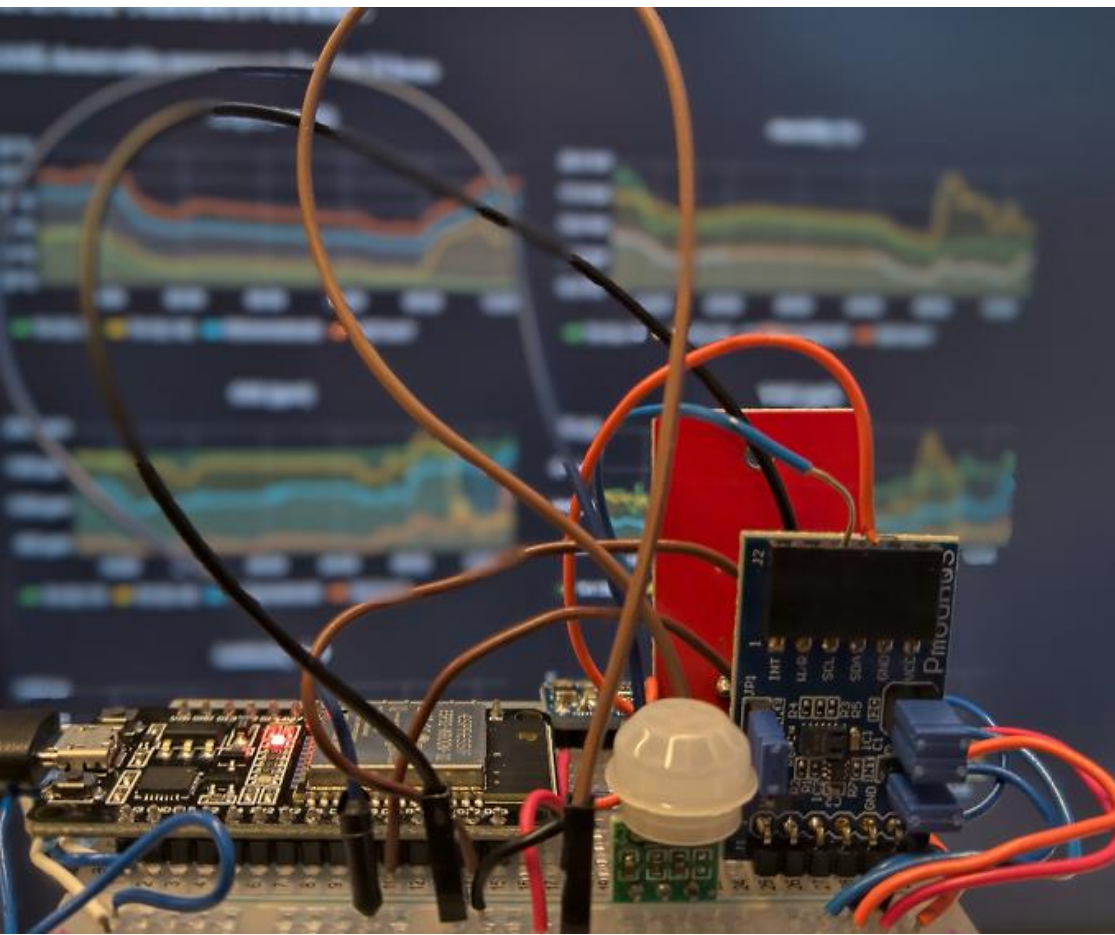


Juha Hauhia

## KAMK-anturiverkko – sensoriverkko

### luokkatilojen monitorointiin



Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2020



**KAMK • University  
of Applied Sciences**

## Tiivistelmä

**Tekijä(t):** Hauhia Juha

**Työn nimi:** KAMK-anturiverkko – sensoriverkko luokkatilojen monitorointiin

**Tutkintonimike:** Insinööri (AMK), tieto- ja viestintätekniikka

**Asiasanat:** sensoriverkot, anturit, langaton tiedonsiirto, esineiden internet, monitorointi, tietokannat

Työn tavoitteena oli kehittää prototyyppi luokkatilojen käyttöasteen seurannan mahdollistavasta sensoriverkosta. Sensoriverkon prototyyppi kehitettiin työn toimeksiantajan Kajaanin Ammattikorkeakoulu Oy:n tiloihin. KAMK-anturiverkon suunnittelussa käytiin läpi sensoriverkon vaatimat osa-alueet keskittyen samalla projektin kustannuksiin. Tavoitteena oli käyttää mahdollisimman paljon avoimen lähdekoodin ratkaisuja.

IoT (Internet of Things) eli esineiden internet on nopeasti kehittyvä teknologian ala. 2000-luvulla suuria harppauksia ottanut langaton tiedonsiirto on mahdollistanut entistä pienempien laitteiden liittämisen verkkoon. Sen myötä esineiden internet on läsnä lähes joka paikassa. IoT-verkkoteknologioita on tänä päivänä tarjolla laaja joukko. Osa verkkoteknologioista sopii parhaiten vähävirtaisiin ja pienen dataliikenteen laitteisiin, kun taas toinen ääripää mahdollistaa suurien datamäärien siirtämisen. Verkkoteknologian valintaan vaikuttaa paljon myös käytettävän kehitysalustan valinta. Pienimmät kehitysalustat ovat jopa kolikkoa pienempiä, mutta laajempia ominaisuuksia hakiessa on yleensä valittava hieman suurempi vaihtoehto. Sensorit ja verkko eivät itsessään vielä muodosta kokonaisuutta. Yleensä sensoriverkon pohjalta löytyy palvelin, joka vastaa datan talteenotosta. IoT-ratkaisuihin sopii yleensä parhaiten aikasarjatyyppinen tietokanta.

KAMK-anturiverkko rakennettiin kahdessa vaiheessa. Ensimmäisenä vuorossa oli verkon infrastruktuurin suunnittelu, jossa käytettäväksi verkkoteknologiaksi valittiin Kajaanin ammattikorkeakoululta jo valmiiksi löytyvä WLAN-verkko. Datan talteenottoa varten asennettiin tietokantapalvelin DC-laboratorion konesalin VMware-virtualisointiympäristöön. Seuraavaksi suunniteltiin sensoryyksikön prototyyppi ESP32-kehitysalustan päälle. Sensoreiden ja kehitysalustan valinnassa huomioitiin kustannukset ja suosittiin avoimen lähdekoodin ratkaisuja. Lisäksi sensoryyksikölle tehtiin prototyyppikotelointi.

Tulevaisuudessa sensoryyksikön jatkokehitys tarjoaa oivan kohteen insinöörikoulutuksen projektiopintoihin. Piirilevyn ja koteloinnin suunnittelu mahdollistaa poikkialaisen projektitoiminnan tieto- ja konetekniikan opiskelijoille. Kun sensoryyksikkö saadaan loppukehitettyä toimivaksi tuotteeksi, voidaan sensoriverkkoa laajentaa koko kampuksen tiloihin opiskelijavoimin. Tulevaisuudessa KAMK-anturiverkkoa voi mahdollisesti myös hyödyntää taloautomaation ohjaamisessa.

Työn laajuudesta huolimatta kokonaisuus oli onnistunut. Työssä pääpainona ollut infrastruktuurin rakennus onnistui ongelmitta, minkä takia sensoryyksikön prototyypin kehitykselle jäi enemmän aikaa. Täten saatiin toimiva esimerkki luokkatilojen monitoroinnissa käytettävästä sensoriverkosta.

## **Abstract**

**Author(s):** Hauhia Juha

**Title of the Publication:** KAMK-anturiverkko – Sensor Network for Classroom Monitoring

**Degree Title:** Bachelor of Engineering, Information and Communication Technology

**Keywords:** sensor networks, sensors, wireless data transmission, internet of things, monitoring, databases

The goal of this bachelor's thesis was to build a prototype of classroom monitoring sensor network. The work was commissioned by Kajaanin Ammattikorkeakoulu Ltd, on whose classrooms the sensor network was to be built. The planning of the KAMK-anturiverkko went through the areas required by the sensor network while keeping the focus on the costs. The aim was to use as many open source solutions as possible.

IoT (Internet of Things) is a rapidly developing field of technology. Wireless communication, which has taken big leaps in the 21<sup>st</sup> century, has made it possible to connect even the smallest devices to the network. With it, the Internet of Things is present in almost every place. Today there is a wide spectrum of IoT network technologies and development platforms. Some of them are best suited for low-power and low-data solutions while the other end allows powerful computations with large amounts of data to be transferred. The choice of development platform and network technology is always made in a way that suits the application. The sensors and the network themselves do not yet form a functioning entity. Usually a sensor network requires a database server responsible for storing the data. A time series database is usually best suited for IoT solutions.

The KAMK-anturiverkko was built in two phases. The first to be designed was a network infrastructure which was built on top of WLAN network already available at Kajaani University of Applied Sciences. A database server was installed in the VMware virtualization environment of the DC laboratory data center. In the second phase, a prototype of the sensor unit was designed for the ESP32 development platform. In both phases, costs were considered, and open source solutions were preferred. Finally, a prototype casing was made for the sensor unit.

In the future, the further development of the sensor unit will serve an excellent target for engineering studies in project studies. The design of the circuit board and enclosure allows cross-disciplinary project activities for students of information and communication technology and mechanical engineering. Once the sensor unit has been developed into a functional product, the sensor network can be expanded to the entire campus with student resources. In the future, the KAMK-anturiverkko may also be used to control building automation.

Although the work was extensive, the outcome was successful. The building of the infrastructure, which was the primary focus of the work, was successful, which left more time for the development of the prototype of the sensor unit. With a properly functioning sensor unit, the work provided a working example of a sensor network used to monitor classrooms.

## Sisällys

1	Johdanto .....	1
2	Suunnittelun lähtökohdat.....	3
2.1	Monitoroitavat tekijät .....	3
2.2	Hinta ja laajennettavuus .....	3
2.3	Kehitysalusta ja tiedonsiirtoprotokollat .....	4
2.4	Palvelin- ja tietokantaratkaisut .....	4
2.5	Datan visualisointi .....	4
3	Internet of Things .....	5
4	Langaton IoT-tiedonsiirto .....	6
4.1	WLAN.....	7
4.2	Bluetooth ja Bluetooth LE .....	8
4.3	6LoWPAN ja ZigBee .....	9
4.4	NB-IoT ja LTE-M .....	10
4.5	Lora ja Sigfox .....	10
4.6	Yhteenveto .....	11
5	IoT-kehitysalustat .....	12
5.1	Raspberry Pi.....	13
5.2	Arduino .....	15
5.3	ESP8266 ja ESP32 .....	16
5.4	Yhteenveto .....	17
6	Palvelin- ja tietokantaratkaisut sekä datan visualisointi .....	18
7	KAMK-anturiverkko: infrastruktuurin rakennus .....	19
7.1	KAMK-anturiverkon palvelin- ja tietokantaratkaisut .....	19
7.1.1	Resurssien määrittäminen.....	19
7.1.2	Virtuaalikoneen laitteisto ja käyttöjärjestelmän asennus .....	20
7.1.3	Tietokannan ja visualisointiympäristön asennus .....	20
7.2	Tiedonsiirtoprotokollan valinta ja verkkoratkaisut .....	24
8	KAMK-anturiverkko: sensoryksikön prototyyppin rakennus .....	26
8.1	Käytettävän kehitysalustan valinta .....	26
8.2	Sensoreiden valinta .....	27

8.2.1	Lämpötila ja ilmankosteus.....	27
8.2.2	Liike .....	28
8.2.3	Hiilidioksidi .....	29
8.2.4	Valoisuus .....	30
8.3	Sensoryksikön kytkennät.....	30
8.4	Sensoryksikön ohjelmointi .....	32
8.4.1	Kehitysalustan lisääminen Arduino Softwareen (IDE).....	32
8.4.2	Tarvittavien ohjelmakirjastojen lisääminen Arduino Softwareen (IDE).....	33
8.4.3	Ohjelmakoodi .....	34
8.5	Sensoryksikön koteloointi .....	35
8.6	Sensoryksikön kokonaiskustannukset .....	37
8.7	Havaitut ongelmat.....	38
9	KAMK-anturiverkko: tulevaisuus .....	39
10	Yhteenveto .....	40
	Lähteet .....	41
	Liitteet	

## Symboliluettelo

3GPP	The 3rd Generation Partnership Project, usean standardointijärjestön yhteistyöorganisaatio
6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks, matalan lähetystehon langaton IPv6 protokolla
A/D-muunnin	Analogia-digitaalimuunnin
AM312	PIR-sensori
Apache2	HTTP-palvelin
Arduino	Avoimeen laitteistoon ja lähdekoodiin perustuva kehitysalue
Arduino Software (IDE)	Avoimeen lähdekoodiin perustuva kehitystyökalu
ARM	Advanced RISC Machines, mikroprosessoriarkkitehtuuri
Bluetooth	Lyhyen kantaman langaton tiedonsiirtotekniikka
Bluetooth LE	Vähävirtainen Bluetooth
Broadcom	Broadcom Corporation, puolijohdepiirivalmistaja
CCS811	Orgaanisia yhdisteitä mittaava sensori
Chronograf	Visualisointityökalu
DHT22	Lämpötilaa ja ilmankosteutta mittaava sensori
ESP8266	Espressif Systemsin IoT käyttöön kehittämä mikrokontrolleri
ESP32	Espressif Systemsin IoT käyttöön kehittämä mikrokontrolleri
GPIO	General Purpose I/O, yleiskäyttöinen pinni mikrokontrollereissa

Grafana	Visualisointityökalu
HTTP	Hypertext Transfer Protocol, protokolla jota WWW-palvelimet käyttävät tiedonsiirtoon
I2C	Yksinkertainen kaksisuuntainen ohjaus- ja tiedonsiirtoväylä
IDII	Interaction Design Institute Ivrea
IEEE	Institute of Electrical and Electronics, kansainvälinen tekniikan alan järjestö
IEEE 802.11	IEEE:n standardi langattomille WLAN-lähiverkoille
InfluxDB	Aikasarjatietokanta
IoT	Internet of Things, eli esineiden Internet
IPsec VPN	IPsec-salausta käyttävä VPN
IPv4	Internet-protokolla, joka toimii internetin perustana
IPv6	IPv4-protokollan paranneltu seuraaja
KamIT	Kajaanin kaupungin eri kouluasteiden yhteinen tietohallintoyksikkö
KAMK	Kajaanin Ammattikorkeakoulu Oy
LDR	Light Dependent Resistor, eli valovastus
Let's Encrypt	Voittoa tavoittelematon taho, joka myöntää sertifikaatteja
Lora	Long Range, vähävirtainen langaton IoT-tietoliikenneverkko
LTE-M	LTE-MTC (Machine Type Communication), LTE-verkossa toimiva IoT-verkkostandardi

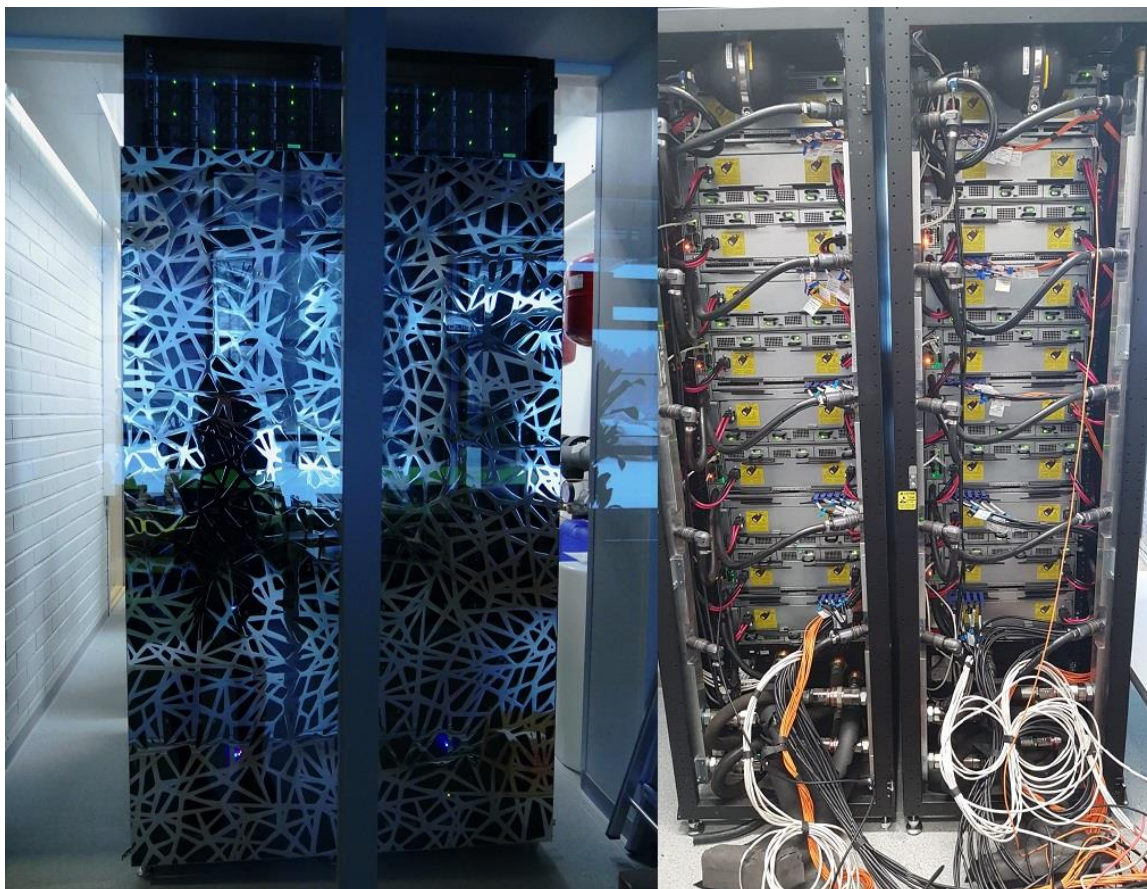
LVM	Looginen taltiohallinta
microSD	Yleisesti digikameroissa ja älypuhelimissa käytetyn SD-muistikortin toistaiseksi pienin versio
NB-IoT	Narrowband Internet of Things, kapeakaistainen IoT-verkko
OpenSSH	Ohjelmistopaketti, joka sisältää Secure Shell (SSH) -työkalut
PIR	Passive Infrared Sensor, infrapunasensori
Raspberry Pi	Luottokortin kokoinen yhden piirilevyn tietokone (puhekielessä: RasPI)
Raspbian	Raspberry Pi-korttitietokoneelle suunniteltu käyttöjärjestelmä
RuuviTag	Lämpöä ja ilmankosteutta mittaava langaton Bluetooth-sensori
Sigfox	Vähävirtainen langaton IoT-tietoliikenneverkko
SPI	Serial Peripheral Interface, sarjamuotoinen oheislaiteväylä
Ubuntu	Debian Linux-jakeluun perustuva käyttöjärjestelmä
VMware	Yhdysvaltalainen ohjelmistoyritys
VPN	Virtual Private Network, virtuaalinen lähiverkko
Wiring	Kehitysalusta, jonka pohjalta Arduino kehitettiin
WLAN	Wireless Local Area Network eli Wi-Fi, langaton lähiverkko
ZigBee	Lyhyen kantaman tietoliikenneverkko



## 1 Johdanto

Kajaanin Ammattikorkeakoulu Oy:ssä (KAMK) elettiin muutosten aikaa. Ammattikorkeakoulun rehtorina vuodesta 2010 toiminut Turo Kilpeläinen luovutti alkuvuodesta 2018 tehtävänsä uuden rehtorin Matti Sarénin hoitoon. Uuden rehtorin työn aloittamista varjostivat valtakunnallisen ammattikorkeakoulu-uudistuksen mukanaan tuomat muutokset ja etenkin alati kiristynvä rahoitus.

Vuonna 2019 alkaneista yt-neuvotteluista huolimatta KAMK:ssa oli käynnissä suuria investointeja. Tieteen tietotekniikan keskus CSC:ltä keväällä 2018 lahjoituksena saadun Bull-supertietokoneen (kuva 1) myötä ammattikorkeakoulun konesalin infrastruktuuriin oltiin tekemässä huomattavia parannuksia. Supertietokonehankkeen lisäksi oli alkamassa uusi koulutusohjelma, Datasta tekoälyyn -insinöörikoulutus (IAMAI), jonka tarkoituksena on tarjota käytännönläheistä koulutusta tekoälyratkaisujen perusteiden parissa. Uuden koulutusohjelman opetuksen perusta on datan kerääminen, hallinta ja analysointi. Edellä mainittujen lisäksi tärkeässä osassa on myös datan tallentamiseen ja laskentaan käytettävän infrastruktuurin ymmärtäminen. [1.][2.]



Kuva 1. Bull-supertietokone edestä (vasen) ja takaa (oikea).

Vaikka ammattikorkeakoulun tiloista löytyy vakuuttava datacenter-ympäristö ja reilusti laskenta-tehoa omaava supertietokone, niin myös analysoitava data on erittäin tärkeässä roolissa. Valmiiksi generoitua datapaketteja on runsaasti saatavilla, mutta käytännönläheistä todellista mittausdataa ei välttämättä olekaan niin helppo saada käsiin. Vaikka todellista mittausdataa saisikin jostain, niin monesti datan lahjoittava taho ei halua, että tuloksia julkistetaan, tai haluaa maksun keräämästään datasta.

Opinnäytetyön tilaajan Kajaanin Ammattikorkeakoulu Oy:n johdon ideana onkin koko koulun kattava sensoriverkko. Sen avulla on tarkoitus kerätä reaaliaikailman dataa kampusalueen rakennusten luokka- ja laboratoriotiloista. Mittausdataa on ensisijaisesti tarkoitus hyödyntää luokkatilojen käyttöasteen seurannassa. Myöhemmin on ehkä mahdollista säätää taloautomaation toimintaa sensoriverkon tuottaman datan perusteella. Samalla itse kerätty data toimii hyvänä analysointikohteena uutta koulutusohjelmaa silmällä pitäen.

Työn tarkoituksena on selvittää sensoriverkon rakentamiseksi tarvittavat hankinnat ja käytettävät verkkoteknologiat. Selvitystyön jälkeen rakennetaan sensoriverkon tarvitsema infrastruktuuri ja prototyyppiratkaisu sensoriyksiköstä. Tärkeänä osa-alueena on mahdollistaa sensoriverkon jatkokehitys osana KAMK:n opetusta. Samalla huomioidaan tarvittavien hankintojen kustannustehokkuutta ja pyritään toteuttamaan sensoriverkko pääosin avoimen lähdekoodin ratkaisuilla.

Ennen teoriaosuutta käydään läpi suunnittelun lähtökohdat. Teoriaosuus on jaettu neljään isompaan osa-alueeseen. Ensin kerrataan lyhyesti, mitä Internet of Things tarkoittaa. Seuraavaksi tutustutaan työn kannalta tärkeisiin verkkoteknologioihin ja kehitysalustoihin. Viimeisenä teoriaosana käsitellään lyhyesti palvelin- ja tietokantaratkaisuja.

Työosuus on jaettu kahteen suurempaan osa-alueeseen. Ensin käydään läpi infrastruktuurin rakennus ja sensoriverkkoon valikoituneet verkkoratkaisut. Toinen osio keskittyy sensoriyksikön prototyyppiratkaisun rakentamiseen. Lopuksi käännetään katse kohti sensoriverkon tulevaisuuden näkymiä ja pohditaan työn herättämiä ajatuksia.

## 2 Suunnittelun lähtökohdat

KAMK-anturiverkon tarkoituksena on kerätä dataa luokkatilojen käyttöasteesta ja ilmanlaadusta. Tässä kohtaa avataan tarkemmin suunnittelussa tärkeimmiksi osa-alueiksi muodostuneita asioita. Koska tarkoituksena on luoda esimerkkiratkaisu koko koulun kattavasta sensoriverkosta, niin työn pääpaino on itse infrastruktuurin suunnittelussa, eikä varsinaisen sensoriyksikön suunnittelussa.

### 2.1 Monitoroitavat tekijät

Pääpainona on luokkatilojen käyttöasteen seuranta. Sensoreiden avulla pyritään seuraamaan, mihin kellonaikoihin luokkatiloissa on ollut ihmisiä. Toinen tärkeänä esille noussut mittaus on valaistus. Sensoriverkon tuottaman datan perusteella pitäisi pystyä selvittämään, onko luokassa ollut ihmisiä ja onko valot päällä. Täten kerätyn datan perusteella voidaan havaita, jos tyhjässä luokkatilassa on ollut valot päällä. Muita tärkeitä perusmittauksia ovat lämpötila, ilmankosteus ja paine-erot muihin rakennuksen tiloihin.

### 2.2 Hinta ja laajennettavuus

Tänä päivänä hinta on esillä monessa asiassa. Sama koskee myös KAMK-anturiverkkoa. Yt-neuvotteluiden myötä entisestään kiristynyt taloustilanne säteilee myös anturiverkon suunnitteluun. Koska sensoriverkkoa on tarkoitus laajentaa ensiksi yhteen rakennukseen (Taito 1) ja sen jälkeen mahdollisesti koko kampusalueelle, on tärkeä huomioitava seikka sensoriyksikön hinta. Koska sensoriverkon laajentaminen tapahtuu pääosin opiskelijavoimin, niin helppo laajennettavuus on tärkeässä roolissa. Sensoriyksikön on oltava helposti käyttöönotettavissa, ja sen pitää rakentua edullisista ja helposti saatavilla olevista komponenteista. Käytön on oltava maksutonta, joten pääpaino on avoimen lähdekoodin ratkaisuissa.

## 2.3 Kehitysalusta ja tiedonsiirtoprotokollat

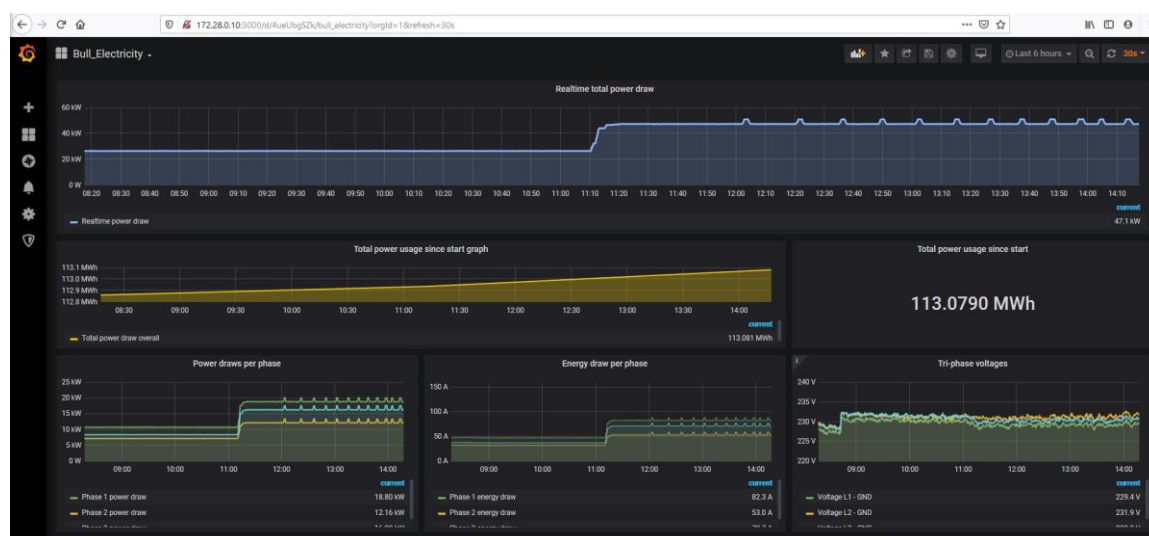
Pyritään löytämään mahdollisimman hyvin tarkoitukseen sopiva kehitysalusta sensoreita varten. Tärkeitä huomioon otettavia seikkoja on sensoriverkon laajennettavuus myös mitattavien tekijöiden osalta. Alustan pitäisi olla avointa lähdekoodia ja mielellään opetuksessa hyödynnettävää teknologiaa. Kartoitetaan käytettävissä olevat verkkoteknologiat, ja pyritään löytämään mahdollisimman hyvin järjestelmään sopiva tiedonsiirtoprotokolla.

## 2.4 Palvelin- ja tietokantaratkaisut

Riittävien palvelin- ja tietokantaratkaisuiden suunnittelu sensoriverkon tuottaman datan talteen ottamiseksi. Tälläkin osa-alueella pääpaino on avoimen lähdekoodin ratkaisuissa. On huomiotava, että kerättyä dataa päästään helposti analysoimaan supertietokoneen avulla. Lisäksi opiskelijoiden olisi hyvä päästä helposti käsiksi dataan.

## 2.5 Datan visualisointi

Kerätyn datan yksinkertainen visualisointi käyttäjälle. Mielekäs ja helppokäyttöinen rajapinta, jonka avulla voi visualisoida dataa eri käyttötilanteisiin. Tarjolla on erilaisia työkaluja kuten Grafana (kuva 2), joiden avulla tietokannassa olevan datan visualisointi on helppoa.



Kuva 2. Grafanalla visualisoitu supertietokoneen virrankulutus

### 3 Internet of Things

Esineiden internet eli Internet of Things (lyhennetään usein kirjainyhdistelmällä IoT) on vanhempi käsite kuin moni uskookaan. Vaikka vasta viime vuosikymmen on tuonut sanan ihmisten huulille, niin käsite esineiden internetistä tulee 90-luvulta, jolloin MIT Auto-ID -laboratorion toiminnanjohtaja Kevin Ashton nimesi sen julkisesti ensimmäistä kertaa. Hyvä esimerkki ensimmäisistä esineistä internetissä on 1980-luvun alkupuolelta: Carnegie Melon yliopistossa paikalliset ohjelmoijat ottivat yhteyttä Coca Cola -automaattiin. Verkon yli tarkastettiin juoman saatavuus sekä se, onko juoma riittävän kylmää. [3.]

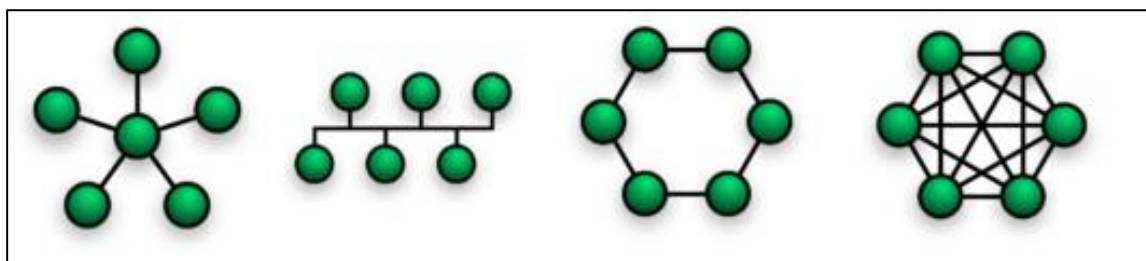
Miksi IoT:stä on tullut päivän sana vasta 2010-luvulla, vaikka ensimmäiset esineet kytkettiin internetiin jo 1980-luvulla? Yksi suuri osatekijä IoT-kehityksessä on tiedonsiirto. 2000-luvulla suuria harppauksia ottanut langaton tiedonsiirto on tuonut verkkoyhteyden entistä pienempiin laitteisiin. Lisääntynyt langattomien verkkoteknologioiden määrä mahdollistaa tarpeen mukaisen kommunikointikeinon erilaisissa käyttökohteissa. Toinen merkittävä vaikuttaja IoT-verkkoihin on 2000-luvun alussa kehitetty IPv6-verkko, joka kehitettiin nykyisen IPv4-verkon seuraajaksi. IoT-verkkojen näkökulmasta tärkein uudistus on kasvanut osoiteavaruus. IPv6-verkossa käytetään 128-bittistä osoiteavaruutta, joka mahdollistaa 340 sekstiljoonaa osoitetta yhdessä verkossa [4]. Suuri osoiteavaruus mahdollistaa sen, että jokainen laite voi yhdistää internetiin omalla osoitteella. Tämä taaskin mahdollistaa kaksisuuntaisen yhteyden laitteeseen.

Vaikka IoT-laitteiden kirjo on varsin suuri, ja niistä puhuttaessa kyse voi olla jopa kokonaisesta rakennuksesta, niin varsin usein pohjalla on jokin mikrokontrolleri, johon on liitetty mittauksen mahdollistavia sensoreita. Nykyään tavallisen kuluttajan saatavilla on lukuisia erilaisia IoT-kehitysalustoja. Ne ovat parhaimmillaan suunniteltaessa prototyyppejä, mutta toimivat varsin hyvin myös lopullisissa ratkaisuissa. Halvimmillaan kehitysalustan hinta on vain muutamia euroja, mutta ominaisuuksien ja suorituskyvyn kasvaessa hinta voi nousta jopa kymmeniin tai satoihin euroihin [5]. Kehitysalustojen ja sensoreiden halpojen hintojen seurausta on se, että IoT-kehitystä tehdään paljon niin sanotuissa DIY-projekteissa (Do It Yourself).

#### 4 Langaton IoT-tiedonsiirto

IoT-laitteen langattomalle kommunikoinnille on lukuisia eri vaihtoehtoja. Laitteessa käytettävä yhteysteknologia valitaan aina käyttökohteen mukaan. Osa tiedonsiirtoprotokollista sopii hyvin suurien datamäärien siirtämiseen, kun taas osa on suunniteltu vähävirtaisuutta ja pientä datamäärää silmällä pitäen. Myös käyttöympäristöllä on väliä. Teollisuuden käyttöön suunnitellut protokollat ovat monesti kalliimpia käyttää, mutta niissä on otettu huomioon teollisuuden tiukat vaatimukset. Tässä osiossa käydään läpi yleisiä langattomia tiedonsiirtoprotokollia, jotka soveltuvat hyvin sisätiloissa käytettäviksi.

Protokollan lisäksi verkkoa suunniteltaessa on otettava huomioon käytettävä verkkotopologia. Topologia määrittää, miten verkon laitteet on liitetty toisiinsa. Langallisissa verkoissa yleisin ratkaisu on tähtitopologia (kuva 3). Muita yleisiä verkkotopologiamalleja ovat väylä-, rengas- ja mesh-topologia (kuva 3).



Kuva 3. Tähti-, väylä-, rengas- ja mesh-topologiat [6].

Tähtitopologiassa kaikki laitteet ovat yhteydessä yhdysliikennepisteeseen, kuten kytkimeen tai langattomaan tukiasemaan. Laitteet eivät siis kommunikoi suoraan toistensa kanssa, vaan kaikki liikenne tapahtuu yhdysliikennepisteen kautta. Väylätopologiassa kaikki laitteet on liitetty yhteen tiedonsiirtokaapeliin t-liittimellä. Rengastopologia muistuttaa hyvin paljon väylätopologiaa, mutta siinä kaapeli on kytketty ympyräksi. Kaksi edellä mainittua ovat kumminkin hyvin herkkiä vikaantumiselle, ja ne ovat vanhentunutta teknologiaa. Mesh-topologiassa jokainen laite on kytketty toisiinsa. Mesh-topologiaa käytetään yleensä langattomissa tiedonsiirtoverkoissa, ja se on hyvin vikasietoinen. [6.]

#### 4.1 WLAN

Puhekielessä käsitteet Wi-Fi ja WLAN (Wireless Local Area Network) eli langaton lähiverkko tarkoittavat monesti samaa asiaa. Todellisuudessa kumminkin WLAN tarkoittaa langatonta lähiverkkoa, ja Wi-Fi on Wi-Fi Alliancen tavaramerkki, joka kattaa joukon langattomia verkkoteknologioita. Wi-Fi Alliancen takana on lukuisia yrityksiä, jotka kehittävät ja valmistavat langattomia verkotuotteita. Vuonna 2017 rikkoutui 800 jäsenyrityksen määrä [7]. Joukossa on monia kaikkien tuntemia isoja yrityksiä, kuten Apple, Samsung, Intel ja Nokia.

IEEE 802.11 -protokolla on Institute of Electrical and Electronics Engineersin (IEEE) standardi langattomille WLAN-verkoille. Protokollan ensimmäinen standardi (802.11-1997) julkaistiin vuonna 1997. Tiedonsiirtonopeus on 1 tai 2 megabittiä sekunnissa ja toiminta vapaalla 2,4 gigahertsin taajuusalueella. Johtuen kasvaneista tiedonsiirtotarpeista jo paria vuotta myöhemmin vuonna 1999 IEEE julkaisi kaksi uutta standardia 802.11a ja 802.11b, jotka eivät kumminkaan olleet yhteensopivia keskenään. 802.11a toimii uudella 5 gigahertsin taajuusalueella, mikä mahdollistaa jopa 54 megabitin tiedonsiirtonopeuden. 802.11b toimii samalla 2,4 gigahertsin taajuudella nostaten tiedonsiirtonopeuden 11 megabittiin sekunnissa. Vuonna 2003 IEEE julkaisi vielä tänäkin päivänä käytetyimmän WLAN-protokollastandardin 802.11g. G-standardi toimii b:n tavoin 2,4 gigahertsin taajuusalueella ja on yhteensopiva vanhemman b-standardin kanssa siirtonopeuden noustessa 54 megabittiin sekunnissa [8]. Seuraavana vuorossa oli 802.11n laajennus vuonna 2009 ja 802.11ac vuonna 2013. 802.11n laajennus nostaa teoreettista tiedonsiirtokykyä 600 megabittiin sekunnissa ollen samalla taajuusyhteensopiva vanhempien a- ja g-standardien kanssa. Yhteensopivuustilassa siirtonopeus on kumminkin sama kuin vanhassa standardissa. Vuonna 2013 julkaistu 802.11ac jatkaa samalla linjalla ollen taaksepäin yhteensopiva b-, g- ja n-standardien kanssa. Tiedonsiirtonopeus kumminkin kasvaa jopa 3466 megabittiin sekunnissa [9]. Tuorein vuonna 2019 julkaistu standardi kantaa nimeä 802.11ax [10]. Se jatkaa 802.11ac:n viitoittamaa tietä ja on myös alaspäin yhteensopiva vanhempien standardien kanssa. Johtuen kalliista hinnasta ja uudesta teknologiasta ax-standardin mukaiset verkot ovat vielä varsin harvinaisia.

Vaikka WLAN ei välttämättä ole ideaalisin verkkoteknologia IoT-ratkaisuihin, niin sen hyvä saatavuus, valmiiksi olemassa olevat tukiasemat ja käytön maksuttomuus tekevät siitä halvan ratkaisun etenkin rakennusten sisällä tapahtuvaan IoT-kommunikointiin. Suuren lähetystehonsa vuoksi sen kuuluvuus esteiden läpi on varsin hyvä (taulukko 1). KAMK-anturiverkon kannalta g- ja n-standardit ovat tärkeimmässä roolissa laitteiden saatavuuden ja valmiin verkon vuoksi.

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
802.11a	5GHz	54Mbit/s	35/120m	star/mesh	0€/laite
802.11b	2.4GHz	11Mbit/s	35/140m	star/mesh	0€/laite
802.11g	2.4GHz	54Mbit/s	38/140m	star/mesh	0€/laite
802.11n	2.4/5.0GHz	600Mbit/s	60/250m	star/mesh	0€/laite
802.11ac	5GHz	6933Mbit/s	35/?m	star/mesh	0€/laite

Taulukko 1. Yleisimpien Wi-Fi standardien perustiedot [11].

## 4.2 Bluetooth ja Bluetooth LE

Bluetoothin kehityksen sysäsi liikkeeseen matkapuhelinvalmistaja Ericsson, kun se alkoi tutkia langattomia tiedonsiirtomenetelmiä matkapuhelimen ja oheislaitteiden välillä. Standardin ylläpidosta vastaava Bluetooth Special Interest Group (Bluetooth SIG) aloitti toimintansa vuonna 1998. Bluetooth 1.0-standardin määrittely julkaistiin vuonna 1999 ja sen paranneltu versio 1.2 vuonna 2003. Tässä vaiheessa tiedonsiirtonopeudet jäivät vielä alle megabitin per sekunti. IoT-laitteiden näkökulmasta suurin uudistus Bluetooth-standardissa tapahtui version 4.0 myötä vuonna 2011 [12]. Sen mukanaan tuoma Bluetooth Low Energy (Bluetooth LE) käyttää samaa 2,4 gigahertsin taajuusaluetta kuin tavallinen Bluetooth, mutta sen virrankäyttö ja tiedonsiirtonopeus on huomattavasti pienempi. Tällä hetkellä uusin Bluetooth-versio on 5.2, joka julkaistiin vuoden 2020 alussa. Bluetooth 5:n suurin uudistus IoT-näkökulmasta on kasvanut kantomatka, minkä mahdollistaa suurempi sallittu lähetysteho [13]. Koska Bluetooth 5-standardi on vasta muutaman vuoden vanha, niin sitä tukevia laitteita on vielä kohtalaisen vähän markkinoilla.

Vähävirtaisuutensa vuoksi Bluetooth LE on houkutteleva valinta IoT-ympäristöihin. Sen käyttö ei vaadi WLAN:in tapaan lisenssimaksuja. Matalan lähetystehon vuoksi se on herkkä esteille, minkä takia kuuluvuus esimerkiksi huoneesta toiseen on huono (taulukko 2). KAMK-anturiverkon kannalta tärkein vertailuun nostettava Bluetooth-standardi kuuluu version 4 alle.

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
Bluetooth 1.0-3.0	2.4GHz	1-3Mbit/s	10/100m	mesh	0€/laite
Bluetooth 4.0 LE	2.4GHz	1Mbit/s	10/100m	mesh	0€/laite
Bluetooth 5.0 LE	2.4GHz	2Mbit/s	40/400m	mesh	0€/laite

Taulukko 2. Bluetooth-standardien perustiedot [13].



### 4.3 6LoWPAN ja ZigBee

6LoWPAN on lyhenne sanoista IPv6 over Low power Wireless Personal Area Networks. Kyse on siis IPv6 matalan tehon langattomista lähiverkoista. Sen standardoinnista vastaa Internet Engineering Task Force (IETF). 6LoWPAN:in tarkoituksena on mahdollistaa kaikista pienimpien ja vähävirtaisimpien laitteiden kommunikointi IP-verkon kautta. Se pohjautuu IEEE 802.15.4 -standardiin, joka on matalan datamäärän ja virran langaton verkkoliikennestandardi. 6LoWPAN toimii usealla taajuusalueella. 2,4 gigahertsin taajuusalue toimii kaikkialla, mutta lisäksi on aluekohtaisia taajuusalueita: 902–928 MHz Pohjois-Amerikassa ja 868–868.6 MHz Euroopassa. [14.]

Saman langattoman verkkoliikennestandardin (IEEE 802.15.4) alla toimiva ZigBee on 6LoWPAN:in varteenotettavin kilpailija. Standardin kehittämisestä vastaa ZigBee-allianssi. Ominaisuuksiltaan se on varsin samanlainen kilpailijaansa verrattuna, mutta se ei käytä IP-pohjaista kommunikointia. Sen seurauksena liitettävyyys IP-verkkoihin on vaikeampaa kuin 6LoWPAN:in tapauksessa. Zigbeeen isona etuna voi pitää nopeata yhteyden muodostamista. [15.]

6LoWPAN ja ZigBee kärsivät pienen lähetystehon vuoksi lyhyestä kantamasta sisätiloissa, joissa on paljon esteitä (taulukko 3). Molempien etuna kumminkin on vähäinen virrankulutus, mikä on suuri etu etenkin paristokäyttöisissä ratkaisuissa. Vaikka 6LoWPAN:in käyttö on ilmaista, niin valmiita infrastruktuuria ei yleensä ole saatavilla, mikä nostaa kustannuksia etenkin käyttöönottaessa. Sama pätee myös ZigBeehen sillä erotuksella, että sen kaupallinen käyttö vaatii ZigBee-allianssiin liittymisen. 6LoWPAN:in etuna ZigBeehen verrattuna voi kumminkin pitää IP-pohjaista liikennöintiä, minkä takia se on helppo sisällyttää muihin IP-pohjaisiin verkkoihin [14].

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
6LoWPAN	2.4GHz	0.25Mbit/s	10/100m	star/mesh	0€/laite
ZigBee	2.4GHz	0.25Mbit/s	10/100m	star/mesh/ cluster	0€/laite

Taulukko 3. 6LoWPAN ja ZigBee-standardin perustiedot [14][15].

#### 4.4 NB-IoT ja LTE-M

NB-IoT (Narrowband Internet of Things) on 3GPP:n (The 3rd Generation Partnership Project) standardisoima verkkoteknologia. Sen tarkoituksena on tarjota hyvä kuuluvuus IoT-laitteille myös sisätiloissa mahdollistaen samalla erittäin matalan virrankulutuksen. NB-IoT käyttää matkapuhelinverkkoa dataliikenteeseen, joten Suomessa se on saatavilla lähestulkoon kaikkialla. Suomalaisista verkko-operaattoreista Telia tarjoaa NB-IoT-aloituspakettia hintaan 450 €/kk. Aloituspaketissa on mukana SIM-kortit viidelle laitteelle 6 kuukauden kokeilujaksoon. Dataliikennettä sallitaan 30 megatavua per sim per kuukausi. NB-IoT käyttää 180 kilohertsin taajuusalueetta ja sen maksimitiedonsiirtonopeus yhteen suuntaan on n. 125 kilobittiä sekunnissa (taulukko 4). [16.]

LTE-M (LTE-MTC [Machine Type Communication]) on nimensä mukaisesti LTE-verkossa toimiva IoT-verkkostandardi. Sen kehityksestä vastaa niin ikään 3GPP. LTE-M:n vahvuuksina NB-IoT:iin verrattuna on nopeampi tietoliikenne (taulukko 4) sekä parempi laitteiden liikuteltavuus [16]. LTE-M toimii 1,08 megahertsin taajuusalueella.

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
NB-IoT	0.2MHz	~0.125Mbit/s	matkapuhelinverkko	star	15€/laite
LTE-M (Cat 1)	1.08MHz	1Mbit/s	matkapuhelinverkko	star	15€/laite

Taulukko 4. NB-IoT ja LTE-M-standardin perustiedot [17].

#### 4.5 Lora ja Sigfox

LoRa (Long Range) on vähävirtainen langaton laajaverkko. Sen kehitti alun perin ranskalainen yhtiö nimeltä Cycleo of Grenoble. Vuonna 2012 yhdysvaltalainen Semtech osti yrityksen ja sen omistaman LoRa-teknologian [18]. Nykyisin LoRa:n kehittämisestä vastaa Semtechin perustama LoRa Alliance ja se on tällä hetkellä ainut radiopiirien valmistaja [19].

Vaikka Semtech valmistaa yksinoikeudella radiopiirit, niin LoRa-laitteiden kommunikointi tapahtuu globaalin avoimen LoRaWAN-protokollan avulla. Taajuusalueena Euroopassa toimii 868 megahertsia, ja verkkoja on tarjolla maksullisten kaupallisten tarjoajien kautta, sekä ei-kaupallisten ilmaisten tarjoajien kautta. LoRaWAN-verkot käyttävät tähtitopologiaa. Matalan taajuuden vuoksi LoRaWAN:in kuuluvuus on hyvä, ja se sietää hyvin esteitä. Tiedonsiirtonopeudet LoRa-laitteilla jäävät varsin pieniksi (taulukko 5). Suomessa LoRa-verkkoa ylläpitää Digita ja verkon käytölle kustannuksia tulee noin 4 €/kk per päätelaite. [20.][21.]

Sigfox toimii LoRa:n tavoin 868 megahertsin taajuusalueella. LoRa:sta poiketen teknologia on avointa, joten laitevalmistajat voivat vapaasti valmistaa verkossa toimivia laitteita. Sigfox on ranskalainen vuonna 2009 perustettu verkko-operaattori. Myös Sigfoxin tiedonsiirtonopeus on hidas (taulukko 5), mutta kuuluvuus on LoRa:n tavoin hyvä. Suomessa Sigfox-verkkoa ylläpitää Connect Finland, ja verkon käyttökustannukset ovat pienimmillään vain euron vuodessa, eli alle 0,1 €/kk. Sigfox-verkossa on kumminkin rajoittavia määräyksiä kommunikoinnille: laite voi lähettää dataa maksimissaan 6 kertaa tunnissa tai 140 kertaa päivässä. [20.]

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
LoRa	868MHz	50kbit/s	5/10+km	star	4€/laite
Sigfox	868MHz	600bit/s	5/10+km	star	0,1€/laite

Taulukko 5. LoRa- ja Sigfox-standardin perustiedot [20].

#### 4.6 Yhteenveto

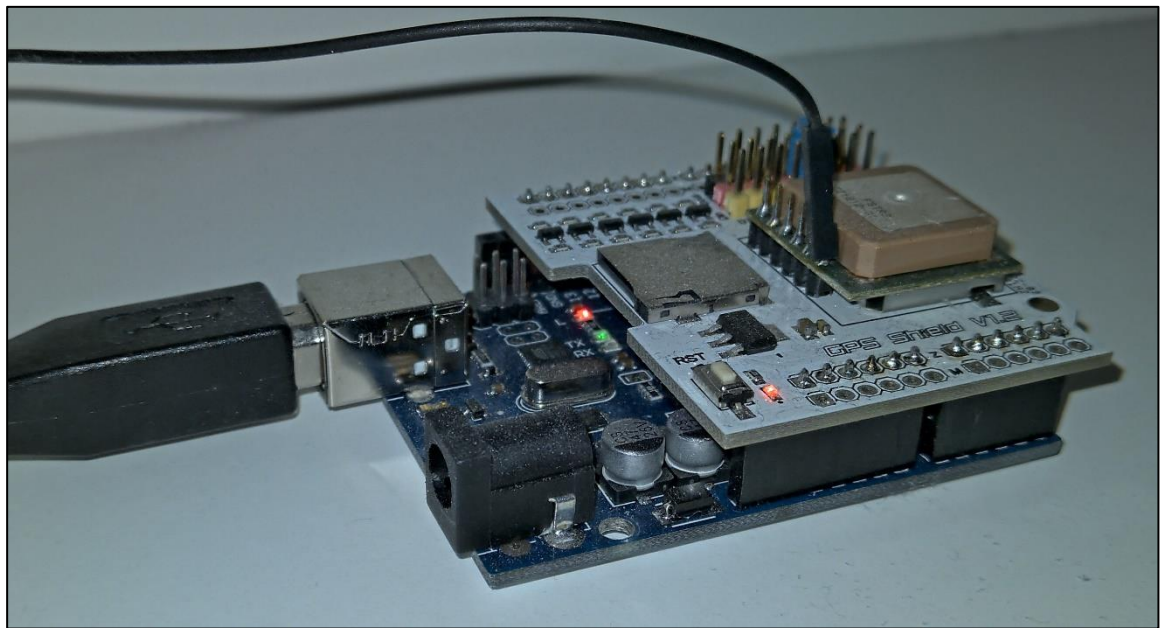
Tänä päivänä IoT-kommunikoinnille on monta vaihtoehtoa. KAMK-anturiverkon näkökulmasta kuukausittainen käyttökustannus (taulukko 6) nousee suureen rooliin, koska monitoroitavia luokka- ja laboratoriotiloja on tilanvarausjärjestelmän mukaan pelkästään Taito 1 -rakennuksessa 38 kappaletta. Tämän takia WLAN nousee jo rakennuksista löytyvän WLAN-infrastruktuurin myötä selvästi varteenotettavimmaksi yhteysteknologiaksi tähän projektiin. WLAN-yhteyden toisena merkittävänä etuna on laitteiden saatavuus. Varsin monessa IoT-kehitysalustassa on sisäänrakennettuna tuki WLAN-yhteydelle.

Nimi	Taajuusalue	Maksimi tiedonsiirtonopeus	Kantama (sisällä/ulkona)	Topologiat	Hinta/kk
802.11n	2.4/5.0GHz	600Mbit/s	60/250m	star/mesh	0€/laite
Bluetooth 4.0 LE	2.4GHz	1Mbit/s	10/100m	mesh	0€/laite
NB-IoT	0.2MHz	~0.125Mbit/s	matkapuhelinverkko	star	15€/laite
LTE-M (Cat 1)	1.08MHz	1Mbit/s	matkapuhelinverkko	star	15€/laite
LoRa	868MHz	50kbit/s	5/10+km	star	4€/laite
Sigfox	868MHz	600bit/s	5/10+km	star	0,1€/laite
6LoWPAN	2.4GHz	0.25Mbit/s	10/100m	star/mesh	0€/laite
ZigBee	2.4GHz	0.25Mbit/s	10/100m	star/mesh	0€/laite

Taulukko 6. Kooste IoT-tiedonsiirtostandardien perustiedoista.

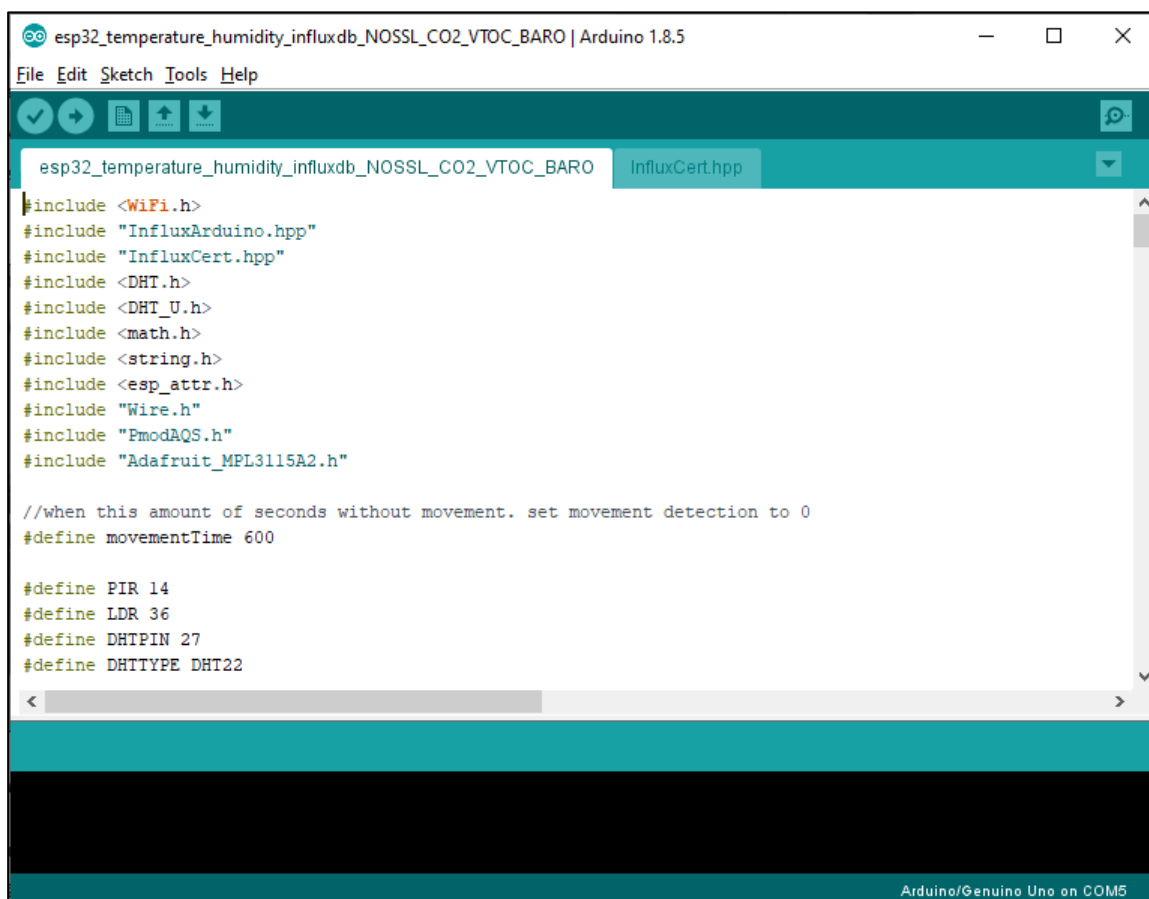
## 5 IoT-kehitysalustat

IoT-laitteen pohjalla on lähestulkoon aina jokin mikrokontrolleri, johon mittauksia tekevät sensorit ja yhteyden muodostamiseen tarvittavat piirit on yhdistetty. Kehitystyötä ja pieniä laitemääriä tehdessä työ tapahtuu yleensä prototyypikehitykseen tarkoitettujen IoT-kehitysalustojen avulla. Pienimmillään ne ovat vain muutamien senttimetrin kokoisia [22], mutta suosituimmat vaihtoehdot ovat lähellä luottokortin kokoluokkaa (kuva 4).



Kuva 4. Arduino Uno -kehitysalusta (sininen), johon kytkettynä GPS-lisäkortti (valkoinen).

Kehitysalustan ohjelmoimiseksi tarvitaan vielä sopiva kehitystyökalu, jolla ohjelmakoodia voidaan kirjoittaa ja ladata kehitysalustalle. Tänä päivänä yksi suosituimmista kehitystyökaluista on Arduino Software (IDE) (kuva 5). Vaikka kilpailijoita löytyy markkinoilta runsain mitoin, niin Arduino Softwaren (IDE) etuna on sen helppokäyttöisyys, avoimen lähdekoodin tuoma yhteensopivuus eri käyttöjärjestelmien kanssa sekä laaja tuki erilaisille Arduino-pohjaisille kehitysalustoille [23]. Avoimen lähdekoodin takia se on myös ilmainen käyttää. Ohjelmistoon voi helposti asentaa kolmannen osapuolen kirjastoja, mikä mahdollistaa myös muiden kuin Arduino-laitteiden ohjelmoinnin Arduino Softwaressa (IDE) [24].



Kuva 5. Arduino Software (IDE)

Tässä osiossa käydään läpi KAMK-anturiverkon suunnittelussa esille nousseita kehitysalustavaihtoehtoja. Koska Arduino Software (IDE) on tällä hetkellä KAMK:n opetuksessa paljon käytetty kehitystyökalu, niin on selvä etu, jos anturiverkon mittausyksikkö on ohjelmoitavissa opiskelijoille tutussa kehitysympäristössä. Tällöin anturiverkon jatkokehitys on helpompi sulauttaa opetukseen.

### 5.1 Raspberry Pi

Raspberry Pi – josta puhekielessä monesti käytetään sanaa ”RasPI” – on muista mukana olevista IoT-kehitysalustoista poiketen noin luottokortin kokoinen yhden piirilevyn tietokone (kuva 6). RasPin kehitys laitettiin alulle Cambridgen yliopiston tietokonelaboratoriossa vuonna 2006. Kehitystyön myötä vuonna 2009 perustettiin Raspberry Pi Foundation, joka julkaisi ensimmäisen Raspberry Pi Model B:nä tunnetun yhden piirilevyn Raspberry Pi -tietokoneen vuonna 2012. Seuraavana vuonna luvassa oli karsituilla ominaisuuksilla varustettu Raspberry Pi Model A. [25.]



Kuva 6. Raspberry Pi Model 3 B +

RasPi:n pohjalla on Broadcomin ARM-pohjainen järjestelmäpiiri, joka sisältää (versiosta riippuen) joko 1 tai 4 prosessoriydintä, grafiikkapiirin ja keskusmuistin. Vuosien myötä uusia Raspberry Pi julkaisuja on tullut tasaiseen tahtiin (taulukko 7). B- ja A-sarjan lisäksi on olemassa vuonna 2015 julkaistu Zero-sarja, joka on kaikista halvin ja karsituin Raspberry Pi -versio. Tämän hetken uusimman ja tehokkaimman mallin Raspberry Pi 4 B:n sydämenä sykkii Broadcomin BCM2711-järjestelmäpiiri, joka tarjoaa neljä 1,5 GHz kellotaajuudella toimivaa ARM v8-prosessoriydintä ja 1, 2 tai 4 gigatavua keskusmuistia. [26.]

Nimi	Virrankulutus	Wi-Fi	Bluetooth	SPI	I2C	GPIO	Hinta/kpl
Raspberry Pi B	700mA	ei	ei	kyllä	kyllä	kyllä	35 €
Raspberry Pi 2 B	800mA	ei	ei	kyllä	kyllä	kyllä	35 €
Raspberry Pi Zero	160mA	ei	ei	kyllä	kyllä	kyllä	5 €
Raspberry Pi 3 B	1340mA	kyllä	4.1 LE	kyllä	kyllä	kyllä	35 €
Raspberry Pi Zero W	180mA	802.11n	4.1	kyllä	kyllä	kyllä	10 €
Raspberry Pi 3 B+	1130mA	802.11 b/g/n/ac	4.2 LE	kyllä	kyllä	kyllä	35 €
Raspberry Pi 4 B	1280mA	802.11 b/g/n/ac	5.0	kyllä	kyllä	kyllä	35 €

Taulukko 7. Raspberry Pi korttitietokoneiden ominaisuuksia [26].

Koska RasPI on itsessään tietokone, niin sen ohjelmointi ei vaadi erillistä kehitystyökalua. Yksittäisen RasPIN kustannuksia kumminkin nostaa tarve microSD-muistikortille, jolle asennetaan käyttöjärjestelmä, sekä USB-virtalähteelle [27]. Varsin usein RasPIssa käytetään Raspbian käyttöjärjestelmää, jonka vaatimuksena on 8 gigatavun microSD-kortti [27].

Raspberry-tuoteperheestä selvästi varteenotettavin vaihtoehto sensoriyksikölle onkin halvan hinnan, matalan virrankulutuksen ja hyvien langattomien yhteyksien vuoksi Zero W. Koska virransyöttö on suunniteltu toteutettavaksi USB-liitännän kautta, niin RasPI ei ole paras vaihtoehto paristokäyttöisissä ratkaisuissa.

## 5.2 Arduino

Arduino-projekti aloitettiin vuonna 2005 Italiassa Interaction Design Institute Ivreassa (IDII), ja sen tarkoituksena oli kehittää halpa ja helppokäyttöinen kehitysalusta niin aloittelijoille kuin edistyneemmillekin kehittäjille. Arduino-projektin pohjana toimii Wiring-kehitysalusta, jonka Hernando Barragán (IDII) kehitti osana ylemmän korkeakoulututkinnon lopputyötä. Koska Arduino-kehitysalustat ovat laitteiston ja ohjelmiston osalta avointa lähdekoodia, niin kuka tahansa voi valmistaa Arduino-laitteita. [28.]

Arduino-tuoteperheeseen kuuluu kymmeniä eri malleja. Niistä ehkä kaikkein tunnetuin on ATmega329-mikrokontrolleriin perustuva Arduino Uno. Se ei ole paras vaihtoehto IoT-kehitysalustaksi, koska siinä ei itsessään ole langattoman tiedonsiirron mahdollistavaa piiriä. Tuoteperheestä kumminkin löytyy lukuisia vaihtoehtoja, jotka on suunniteltu IoT-kehitystä silmällä pitäen, ja ne rakentuvat ARM-pohjaisen järjestelmäpiirin päälle [29]. Alla taulukoituna KAMK-anturiverkon näkökulmasta kaksi kiinnostavinta Arduino IoT-mallia, sekä vertailumallina Arduino Uno WiFi (taulukko 8).

Nimi	Virrankulutus	Wi-Fi	Bluetooth	SPI	I2C	GPIO	Hinta/kpl
Arduino Uno WiFi	<50mA	802.11b/g/n	ei	kyllä	kyllä	kyllä	39 €
Arduino MKR1010 WiFi	<20mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	31 €
Arduino Nano 33 IoT	<20mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	16 €

Taulukko 8. Arduino-kehitysalustojen perustietoja [30, 31, 32].

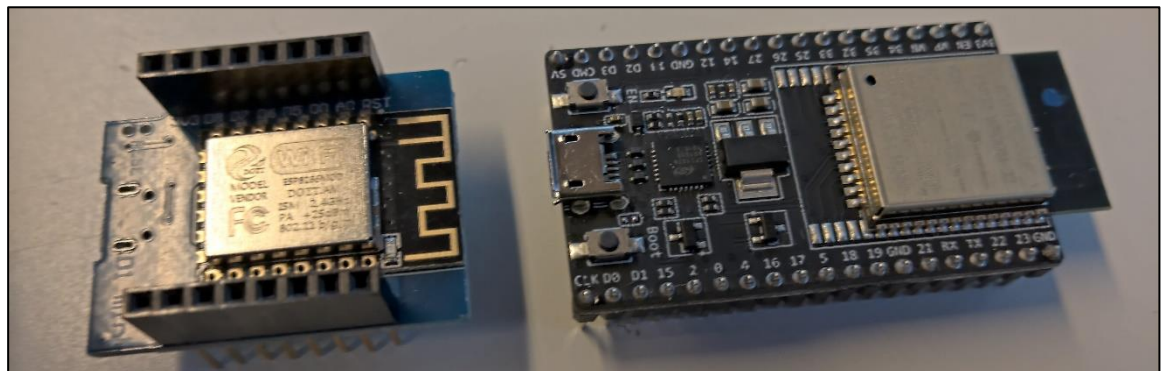
Yllä taulukoiduista malleista etenkin Arduino MKR1010 Wi-Fi on IoT-kehityksen näkökulmasta erittäin kiinnostava malli. Vaikka sen hinta on kohtalaisen korkea, niin taulukon ominaisuuksien



lisäksi se sisältää lataamiseen kykenevän liittimen akulle [30]. Erittäin matalan virrankulutuksen ja hyvien langattomien yhteyksien takia MKR1010 Wi-Fi onkin hyvä vaihtoehto paristokäyttöistä IoT-ratkaisua suunniteltaessa.

### 5.3 ESP8266 ja ESP32

ESP8266 ja sen seuraaja ESP32 (kuva 7) ovat kiinalaisen Espressif Systems -yrityksen kehittämiä edullisia mikrokontrollereita, joissa Wi-Fi ja mikroprosessori ovat integroituna samalle piirille. Yrityksen pääkonttori sijaitsee Shanghaissa, ja ensimmäisen maailmalla laajasti tunnetun IoT-tuotteen ESP8266:n julkaisu sijoittuu vuoteen 2014 [33]. Laitteen läpimurtoa IoT-markkinoilla hidasti aluksi pelkkä kiinankielinen dokumentaatio sekä huono saatavuus Kiinan ulkopuolella [34]. Vuonna 2016 oli luvassa ESP8266:n seuraajan ESP32:n julkaisu. Merkittävimpiä uudistuksia olivat Bluetooth-yhteys sekä toinen mikroprosessoriydin [33].



Kuva 7. ESP8266 (vas.) ja ESP32 (oik.) kehitysalustat

ESP8266 suunniteltiin alun perin Wi-Fi-piiriksi muille mikrokontrollereille. Pian markkinoille tulon jälkeen kumminkin huomattiin, että siinä on enemmän laskentatehoa ja muistia kuin monessa muussa mikrokontrollerissa, joita IoT-kehityksessä käytettiin. Piirin julkaisuvuonna Ivan Grokhotkov julkaisi projektin, jonka tarkoituksena oli tuoda ESP8266:lle tuki Arduino Softwareen (IDE). ESP8266:n ongelmana on sen heikko tietoturva. Puuttuva rautatason salausta mahdollistaa sen, että kuka tahansa, jolla on fyysinen pääsy laitteeseen, voi lukea siitä esimerkiksi Wi-Fi-käyttäjätunnukset. Vuonna 2019 Espressif Systems julkaisi karsitun version ESP32-mikrokontrollerista, joka kantaa nimeä ESP32-S2. Sitä voi pitää ominaisuuksien puolesta paremman tietoturvan tarjoavana korvaajana ESP8266-piirille. [35.]



Nimi	Virrankulutus	Wi-Fi	Bluetooth	SPI	I2C	GPIO	Hinta/kpl
ESP8266 dev board	<30mA	802.11b/g/n	ei	kyllä	kyllä	kyllä	9 €
ESP32 dev board	<30mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	10 €

Taulukko 9. ESP8266:n ja ESP32:n perusominaisuudet [35] (hinnat digikey.com).

Espressif Systemsin piirit ovat halvan hinnan, kattavien ominaisuuksien ja pienen virrankulutuksen ansiosta erittäin varteenotettavia vaihtoehtoja IoT-ratkaisuihin. Koska lähestulkoon kaikki ominaisuudet on rakennettu yhdelle piirille, niin kehitysalustojen hinta on edullinen. ESP32 ja ESP8266 sopivat hyvin myös paristokäyttöisiin IoT-ratkaisuihin. ESP32-S2-piiriin perustuvaa valmista kehitysalustaa ei työn suorittamisen aikaan ollut vielä saatavilla.

#### 5.4 Yhteenveto

Erilaisia IoT-kehitysalustoja on olemassa sadoittain, ja eri vaihtoehtoista voisi kirjoittaa vaikka kirjan. KAMK-anturiverkon tapauksessa keskitytään pääasiassa jo valmiiksi KAMK:lta löytyviin tai nopealla aikataululla saatavissa oleviin ratkaisuihin (taulukko 10). Ammattikorkeakoululta jo valmiiksi löytyvän WLAN-infrastruktuurin vuoksi katsotaan WLAN-yhteyden olevan yksi välttämättömistä ominaisuuksista. Koska työn pääasiallisena tarkoituksena on rakentaa infrastruktuuri KAMK-anturiverkolle, niin kehitysalustan valinnassa halutaan jättää mahdollisimman paljon vaihtoehtoja erilaisille sensori- ja virransyöttöratkaisuille.

Nimi	Virrankulutus	Wi-Fi	Bluetooth	SPI	I2C	GPIO	Hinta/kpl
Raspberry Pi B	700mA	ei	ei	kyllä	kyllä	kyllä	35 €
Raspberry Pi 2 B	800mA	ei	ei	kyllä	kyllä	kyllä	35 €
Raspberry Pi Zero	160mA	ei	ei	kyllä	kyllä	kyllä	5 €
Raspberry Pi 3 B	1340mA	kyllä	4.1 LE	kyllä	kyllä	kyllä	35 €
Raspberry Pi Zero W	180mA	802.11n	4.1	kyllä	kyllä	kyllä	10 €
Raspberry Pi 3 B+	1130mA	802.11 b/g/n/ac	4.2 LE	kyllä	kyllä	kyllä	35 €
Raspberry Pi 4 B	1280mA	802.11 b/g/n/ac	5.0	kyllä	kyllä	kyllä	35 €
Arduino Uno WiFi	<50mA	802.11b/g/n	ei	kyllä	kyllä	kyllä	39 €
Arduino MKR1010 WiFi	<20mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	31 €
Arduino Nano 33 IoT	<20mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	16 €
ESP8266 dev board	<30mA	802.11b/g/n	ei	kyllä	kyllä	kyllä	9 €
ESP32 dev board	<30mA	802.11 b/g/n	4.2 LE	kyllä	kyllä	kyllä	10 €

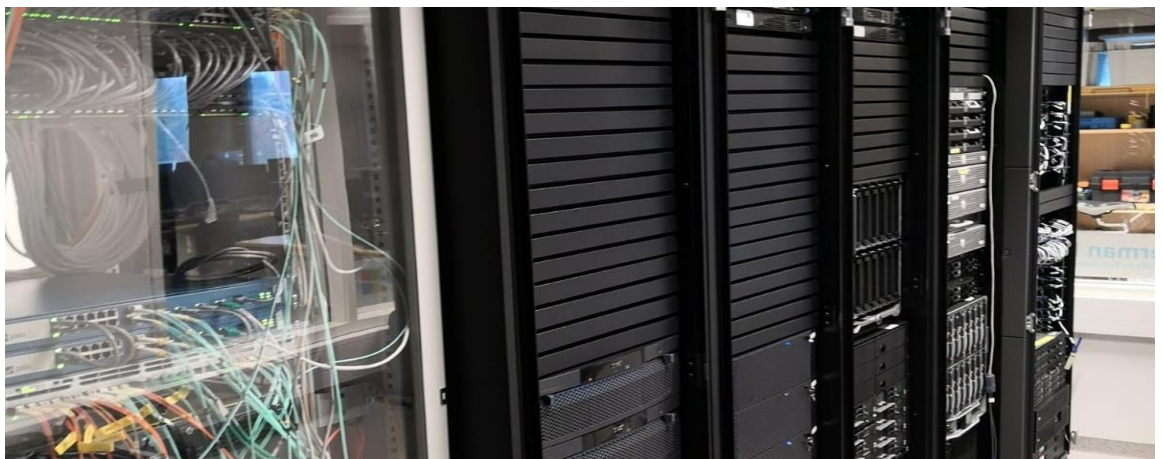
Taulukko 10. Kooste IoT-kehitysalustojen perustiedoista.

## 6 Palvelin- ja tietokantaratkaisut sekä datan visualisointi

Sensoridatan tallentamiseksi tarvitaan tietokanta, johon data tallennetaan. Koska sensorien tuottama data on aikasarjatyypistä, niin paras vaihtoehto datan tallentamiseksi on datatyypille optimoitu tietokanta [36]. DB-engines-sivusto listaa erilaisia tietokantatyyppejä ja asettaa ne paremmuusjärjestykseen suosion mukaan [37]. KAMK-anturiverkon näkökulmasta on järkevintä löytää mahdollisimman suosittu ja hyvin tuettu tietokantaratkaisu. Yleensä suosituille ratkaisuille on olemassa kattavasti valmiita kirjastoja, joiden avulla data saadaan helposti siirrettyä sensorilta tietokantaan.

Time series -tyyppisistä tietokannoista InfluxDB on selvästi suosituin. Se on avointa lähdekoodia ja rakennettu etenkin IoT-ratkaisuiden tuottamaa massiivista datamäärää silmällä pitäen. InfluxDB:n etuna on hyvä tuki datan analysoimiselle ja visualisoinnille. Tuettujen visualisointityökalujen listalta löytyy muun muassa suosittuja työkaluja kuten Chronograf ja Grafana. InfluxDB tukee suurta määrää samanaikaisia kirjoituksia, ja siinä on hyvät sisäänrakennetut ominaisuudet datan pakkaamiseksi ja turhan datan poistamiseksi tietokannasta. [38.]

Tietokanta tarvitsee alleen palvelimen, jolle se asennetaan. Palvelimen on oltava aina verkossa, jotta sensorit voivat lähettää dataa palvelimella käynnissä olevaan tietokantaan. Yleensä palvelin ostetaan joltakin konesalipalveluntarjoajalta, mutta koska KAMK:n tiloista löytyy datacenter-koulutuksen ytimenä toimiva konesali (kuva 8), jonka laitteisto riittää helposti yhden tietokantapalvelimen ylläpitämiseen, niin tässä tapauksessa palvelinratkaisut voidaan hankkia talon sisältä.



Kuva 8. KAMK datacenter-laboratorion konesalin räkkikaapit

## 7 KAMK-anturiverkko: infrastruktuurin rakennus

Työn ensimmäisessä vaiheessa rakennetaan infrastruktuuri sensoriverkkoa varten. Palvelimen sijainnille vaihtoehtoina ovat CSC-konesali Renforsin rannan yritysalueella ja DC-laboratorion konesali. Koska KAMK-anturiverkon jatkokehitystä on tarkoitus toteuttaa opetuksen parissa, niin loogisin valinta palvelimen sijainnille on KAMK:n omissa tiloissa sijaitseva DC-laboratorion konesali. Tässä vaiheessa rekisteröidään myös anturiverkko.fi-verkkotunnus Joker.com-verkkotunnus-palvelusta.

### 7.1 KAMK-anturiverkon palvelin- ja tietokantaratkaisut

Palvelimen sijainniksi valitaan DC-laboratorion konesalin VMware-virtualisointiympäristö. Virtualisoinnilla tarkoitetaan sitä, että virtuaalinen tietokone pyörii jonkin oikeaa laitteistoa sisältävän järjestelmän sisässä [39]. Tällöin virtuaalisen tietokoneen laitteisto on erotettuna isäntäjärjestelmästä, ja virtuaalikoneen käyttäjälle ovat näkyvissä vain virtuaalikoneelle määrätyt resurssit [39]. Virtualisoinnin etuna on se, että saman isäntäkoneen resursseja voidaan käyttää useamman toisistaan erillään olevan palvelun ajamiseen [39]. Jotta virtuaalikoneen resurssit voidaan määrittää oikein, on ennen palvelimen luomista ja asentamista selvitettävä, mitä käyttöjärjestelmää ja palveluita sillä tullaan ajamaan. Vaikka liian suuret resurssit eivät yleensä haittaa itse virtuaalikoneen toimintaa, niin se varaa turhaan resursseja järjestelmän muilta koneilta. Lisäksi resurssien lisääminen on yleensä paljon ongelmattomampaa kuin niiden vähentäminen. Etenkin vikatilanteen sattuessa pienemmillä resursseilla varustettu virtuaalikone on helpompi siirtää toiseen järjestelmään. Siksi järjestelmän tehontarve on hyvä selvittää mahdollisimman tarkasti jo suunnitteluvaiheessa.

#### 7.1.1 Resurssien määrittäminen

Tietokannaksi valitaan influxDB-aikasarjatietokanta. InfluxDB:n virallisesti tuettuihin käyttöjärjestelmiin kuuluu useita Linux-jakeluita [40]. Tähän työhön käyttöjärjestelmäksi valikoituu työn tekijälle jo muista projekteista tutuksi tullut Ubuntu 18.04.2 LTS. Käyttöjärjestelmän valintaa puoltaa influxDB dokumentaatiosta suoraan Ubuntu-jakelulle löytyvät ohjeet. Toinen käyttöjärjestelmän

valintaan vaikuttava tekijä on Grafana-visualisointiympäristön tuki Ubuntuille [41]. InfluxDB:n dokumentaation mukaan alle 5000 kirjoitusta ja alle 5 lukua per sekunti palveleva tietokanta tarvitsee 2–4 prosessoriydintä ja 2–4 gigatavua keskusmuistia [42]. Levytilan tarpeen määrittämisessä varmin tapa on luoda testidataa riittävän pitkältä ajanjaksolta ja seurata tietokannan koon kasvua. Koska tässä vaiheessa ei ole vielä täyttä varmuutta kaikista monitoroitavista tekijöistä, niin levytilaa varataan kohtuudella ja päätetään tarkastaa tilantarve, kun sensoridataa on saatu pidemmältä ajanjaksolta tietokantaan.

### 7.1.2 Virtuaalikoneen laitteisto ja käyttöjärjestelmän asennus

Virtuaalikoneen resurssit määritetään pääasiallisesti InfluxDB:n vaatimusten mukaan. Keskusmuistin määrä tuplataan minimivaatimuksista, koska suurempi muistin määrä nopeuttaa InfluxDB:n tietokantahakuja [42].

Virtuaalikoneen laitteisto VMware-virtualisointiympäristössä:

- Prosessoriydinten lukumäärä: 2
- Keskusmuististin määrä: 8 gigatavua
- Levytilan määrä: 200 gigatavua (thin provision)

Asennusmediaa käytetään Ubuntu server 18.04.2 LTS liveä. Asennuksessa isäntänimeksi (hostname) määritetään anturiverkko, verkkokortin IP-osoite määritetään DC-laboratorion verkon vaatimusten mukaisesti ja levy alustetaan käyttäen loogista taltiohallintaa (LVM). LVM mahdollistaa tallennustilan varaamisen joustavammin kuin perinteiset osiot. Täten levytilan kasvattaminen myöhemmin on helpompaa. Lisäksi luodaan käyttäjätunnus ja asennetaan OpenSSH-etäyhteyttä varten. Muut asetukset jätetään oletusarvoon.

### 7.1.3 Tietokannan ja visualisointiympäristön asennus

Käyttöjärjestelmän asennuksen ja virtuaalikoneen uudelleenkäynnistyksen jälkeen kirjaututaan palvelimelle SSH-yhteydellä ja varmistetaan, että paketinhallinta on ajan tasalla sekä kaikki paketit ovat uusimmassa saatavilla olevassa versiossa (koodiesimerkki 1).

```
sudo apt-get update
sudo apt-get upgrade
```

### Koodiesimerkki 1. Päivityskomennot

Seuraavaksi asennetaan InfluxDB ja Grafana niiden omien dokumentaatioiden mukaisia komentoja käyttäen (koodiesimerkki 2).

```
wget -qO- https://repos.influxdata.com/influxdb.key |sudo apt-key add -
source /etc/lsb-release
echo "deb https://repos.influxdata.com/ubuntu bionic stable" |sudo tee
/etc/apt/sources.list.d/influxdb.list
sudo apt-get update
sudo apt-get install influxdb
sudo apt-get install -y software-properties-common
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb
stable main"
sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key
add -
sudo apt-get update
sudo apt-get install grafana
```

### Koodiesimerkki 2. InfluxDB:n ja Grafanan asennuskomennot [43, 44].

Asennuksen jälkeen muokataan InfluxDB:n asetustiedostosta autentikointi päälle (koodiesimerkki 3). Tällöin tietokanta vastaa vain oikeilla käyttäjätunnuksilla lähetettyihin HTTP-pyyntöihin.

```
#/etc/influxdb/influxdb.conf
[http]
# Determines whether user authentication is enabled over HTTP/HTTPS.
auth-enabled = true
```

### Koodiesimerkki 3. InfluxDB:n autentikointiasetus

Asetusten muokkaamisen jälkeen käynnistetään tietokanta ja InfluxDB komentoliittymä (koodiesimerkki 4).

```
sudo systemctl start influxdb
influx
```

### Koodiesimerkki 4. InfluxDB käynnistys sekä komentoliittymän käynnistäminen

Koska autentikointi asetetaan päälle, niin InfluxDB vaatii käyttäjää luomaan pääkäyttäjän. Lisäksi luodaan käyttäjätunnus ja tietokanta KAMK-anturiverkkoa varten (koodiesimerkki 5).

```

Connected to http://localhost:8086 version 1.7.10
InfluxDB shell version: 1.7.10
> CREATE USER admin WITH PASSWORD '<password>' WITH ALL PRIVILEGES
> auth
username: admin
password: <password>
> CREATE DATABASE anturiverkko
> CREATE DATABASE anturiverkko
> CREATE USER "anturiverkko" WITH PASSWORD '<password>'
> GRANT ALL ON anturiverkko TO anturiverkko

```

**Koodiesimerkki 5.** InfluxDB komennot tietokannan ja käyttäjien luomiseksi.

InfluxDB-aikasarjatietokannan asentamisen jälkeen siirrytään muokkaamaan Grafana-visualisointiympäristön asetuksia. Grafanassa käyttäjien oikeudet määritetään organisaatioihin. Oletusorganisaatio ei salli anonyymiä graafien selaamista, joten Grafanaan luodaan uusi organisaatio julkista kaikkien luettavissa olevaa dataa varten. Ennen organisaation luomista Grafanan asetuksiin määritetään luotavalle organisaatiolle anonyymien luvun salliva rivi (koodiesimerkki 6). Lisäksi rekisteröidään alidomain grafana.anturiverkko.fi Joker.com-verkkotunnuspalvelusta ja asetetaan se Grafanan julkiseksi domainiksi.

```

# /etc/grafana/grafana.ini
[server]
# The public facing domain name used to access grafana from a browser
domain = grafana.anturiverkko.fi
[auth.anonymous]
# enable anonymous access
enabled = true
# specify organization name that should be used for unauthenticated
#users
org_name = Anonyymit-apinat
# specify role for unauthenticated users
org_role = Viewer

```

**Koodiesimerkki 6.** Grafana asetusten editointi anonyymien luvun sallimiseksi

Asetustiedoston muokkaamisen jälkeen käynnistetään Grafana komentoriviltä (koodiesimerkki 7).

```

sudo systemctl start grafana-server

```

**Koodiesimerkki 7.** Grafanan käynnistys

Lopuksi käännetään katse kohti tietoturva. Grafana-visualisointiympäristön avulla tehtyjä graafeja halutaan päästä lukemaan ja muokkaamaan julkisen verkon puolelta. Tietoturvan kannalta on tärkeää, että kirjautumiseen käytettäviä tunnuksia ja salasanoja ei lähetetä selkokielenä verkon yli. Grafanassa itsessään on tuki salausta varten, mutta salauksen voi toteuttaa myös

etupalvelimen (proxy) avulla. Tällöin asiakasohjelma kommunikoi salattuna etupalvelimen kanssa, joka välittää liikenteen Grafana-visualisointiympäristöön.

Sertifikaatti päätetään hankkia ilmaiseksi Let's Encrypt -palvelun kautta. Let's Encryptin tarjoamat sertifikaatit vanhenevat kolmessa kuukaudessa, jos niitä ei päivitä. Joker.com-verkkotunnuspalvelu tukee Let's Encrypt -sertifikaatteja, ja sertifikaattien päivittäminen toteutettiin Joker.com dokumentaation mukaisesti dehydrated-työkalun avulla [45]. Linuxin crontabia apuna käyttäen asetetaan sertifikaatin päivityskomento ajoon viikoittaisesti luomalla komennot sisältävä tiedosto /etc/cron.weekly/ -hakemistoon.

HTTP-palvelimeksi asennetaan Ubuntun paketinhallinnasta Apache2 HTTP -palvelin (koodiesimerkki 8).

```
sudo apt-get install apache2
```

#### Koodiesimerkki 8. Apache2 HTTP -palvelimen asennus

Asennuksen jälkeen muokataan Apachen asetustiedostoa ja lisätään sinne nimipohjainen virtuaalipalvelu Grafanaa varten (koodiesimerkki 9).

```
#/etc/apache2/sites-enabled/default-ssl.conf
<VirtualHost *:443>
    ServerName grafana.anturiverkko.fi
    ServerAdmin admin@anturiverkko.fi

    Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload"
    Header always set Referrer-Policy "no-referrer"

    SSLCertificateFile
/var/lib/dehydrated/certs/anturiverkko.fi/fullchain.pem
    SSLCertificateKeyFile
/var/lib/dehydrated/certs/anturiverkko.fi/privkey.pem

    SSLEngine on
    SSLProtocol all -TLSv1 -TLSv1.1 -SSLv2 -SSLv3
    SSLHonorCipherOrder on
    SSLCompression off
    SSLOptions +StrictRequire
    SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-
POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-
AES128-SHA256:ECDHE-RSA-AES128-SHA256
    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:3000/
    ProxyPassReverse / http://127.0.0.1:3000/
</VirtualHost>
```

#### Koodiesimerkki 9. Apache2-asetustiedoston muutokset

Muokkausten jälkeen käynnistetään Apache2 komentoriviltä (koodiesimerkki 10).

```
sudo systemctl start apache2
```

Koodiesimerkki 10. Apache2-palvelimen käynnistyskomento

Apache2 vaatii sertifikaatin muuttuessa uudelleenkäynnistyksen. Sertifikaattien päivitysten vuoksi lisätään Apachen uudelleenkäynnistävä komento (`sudo systemctl restart apache2`) jo aiemmin luodun sertifikaattien päivityksen ajastavan tehtävän loppuun.

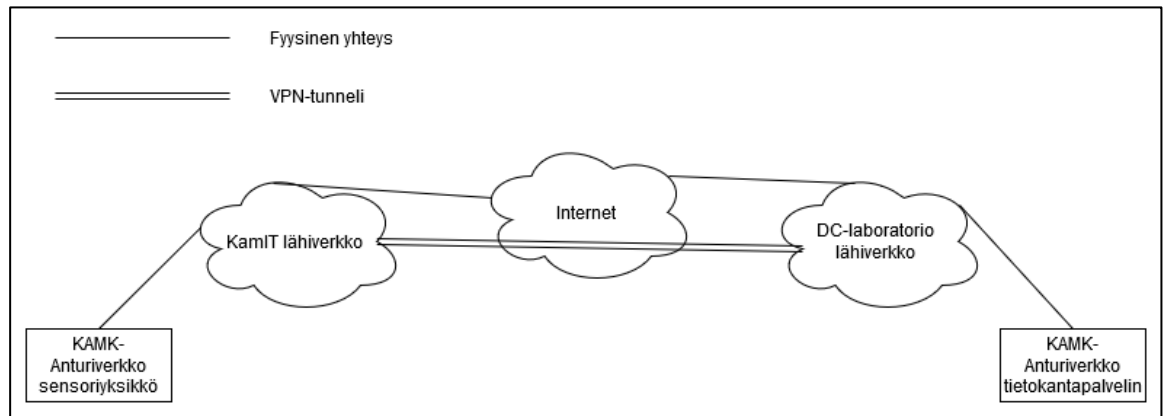
Asennusten ja palveluiden käynnistämisen jälkeen kirjaututaan Grafanaan osoitteessa <https://grafana.anturiverkko.fi> ja havaitaan, että kirjautuminen onnistuu ongelmitta. Onnistuneen sisäänkirjautumisen jälkeen lisätään InfluxDB tietolähteeksi molempiin organisaatioihin ”Configuration” -sivun ”Data Sources” -osiosta. Palvelimen osoitteeksi määritetään localhost ja tietokannan nimi, sekä kirjautumistiedot aiemmin luotujen tietojen mukaan. ”Save & Test” -painikkeen painamisen jälkeen visualisointiympäristö on valmis.

## 7.2 Tiedonsiirtoprotokollan valinta ja verkkoratkaisut

Aiempien IoT-projektien myötä on tiedossa, että KamIT Tietohallinnon ylläpitämään koko KAMK:n kattavaan WLAN-verkkoon on mahdollista luoda oma verkko IoT-laitteita varten. Tässä vaiheessa ongelmana on sensoriyksikön salaamaton kommunikointi KAMK-anturiverkko-palvelimen kanssa. Salattu liikenne sensoriyksikön ja palvelimen välillä on täysin mahdollista, mutta se vaatii enemmän ominaisuuksia ja tehoa sensoriyksikön mikroprosessorilta. Sensoriverkon data ei itsessään ole salaista, mutta koska tulevaisuudessa dataa on tarkoitus käyttää taloautomaation ohjaamiseen, niin on hyvä ottaa tietoturva huomioon. Tällöin datan manipuloiminen ja sen myötä mahdollisen vahingon aiheuttaminen on hyökkääjän näkökulmasta huomattavasti haastavampaa.

Verkotuksen toteuttamiseksi pidettiin palaveri yhdessä KamIT Tietohallinnon ja DC-laboratorion ylläpidon kanssa. Sovittiin, että IoT-laitteita varten luodaan langaton lähiverkko KAMK:n kampusalueelle. Samassa palaverissa selvisi, että on mahdollista muodostaa IPsec VPN (virtuaalinen lähiverkko) DC-laboratorion ja KamITin verkkojen välille (kuva 9). IPsec VPN tarkoittaa sitä, että kahden lähiverkon välille muodostetaan salattu tunneli, jonka läpi verkot kommunikoivat keskenään [46]. Tällöin tunnelin molemmissa päissä verkkoon liitetyt laitteet ovat virtuaalisesti samassa lähiverkossa, vaikka fyysinen yhteys kulkee julkisen internetin kautta [46].





Kuva 9. KamIT lähiverkon ja DC-laboratorion lähiverkon välinen IPsec VPN

Verkkoratkaisuiden myötä infrastruktuuri KAMK-anturiverkkoon saadaan valmiiksi. KamIT Tietohallinnon ja DC-laboratorion välinen VPN-tunneli ratkaisee ongelman tietokannan kirjautumistietojen salaamiseksi. Luotujen verkkoratkaisuiden myötä sensoriyksiköt ja tietokantapalvelin toimivat näennäisesti samassa lähiverkossa.

## 8 KAMK-anturiverkko: sensoriyksikön prototyypin rakennus

Toimivan infrastruktuurin rakentamisen jälkeen kokonaisuudesta puuttuu enää sensoriyksikkö. Työn laajuudesta ja aikataulullisista syistä päätetään, että viimeistellyn sensoriyksikön sijaan lähdetään kehittämään prototyyppiratkaisua, jonka jatkokehitys otetaan osaksi opetusta.

### 8.1 Käytettävän kehitysalustan valinta

Ensimmäistä prototyyppiratkaisua lähdetään rakentamaan Raspberry Pi Model 3 b+:n päälle. Ensimmäinen sensoriyksikköön liitetty sensori on Bluetoothilla toimiva lämpöä ja ilmankosteutta mittaava RuuviTag. Tässä vaiheessa ajatuksena on tehdä RasPista jokaiseen huonetilaan reititin, johon sensorit liitetään Bluetoothilla. Ratkaisun etuna on sensoreiden helppo sijoittaminen mittauksen kannalta parhaaseen paikkaan. Valitettavasti sensoriyksikkökohtaiset kustannukset nousevat tässä ratkaisussa liian suuriksi. Pelkkä lämmön ja ilmankosteuden mittaus RasPI-pohjaisessa ratkaisussa nostaa sensoriyksikön hinnan yli 100 euroon. Ratkaisulla päästään kumminkin testaamaan infrastruktuurin toimivuus, ja dataa saadaan lähetettyä tietokantaan onnistuneesti. Samalla havaitaan, että visualisointiympäristö toimii halutulla tavalla. Tässä vaiheessa kehitysalustan valinnan suhteen päätetään kumminkin palata suunnittelupöydän ääreen ja etsiä kustannustehokkaampi vaihtoehto.

Kehitysalustan ohjelmointia silmällä pitäen käydään läpi vaihtoehtoja, joiden ohjelmakoodia voi kehittää Arduino Software (IDE) -kehitystyökalulla. Ensimmäinen vakavaan harkintaan päässyt kehitysalusta on jo valmiiksi KAMK:n laboratorioista löytyvä ESP8266. Prototyyppiratkaisuun halutaan kumminkin sisällyttää mahdollisuus liittää Bluetoothilla kommunikoivia sensoreita. Tässä vaiheessa katse käännetään kohti ESP32-kehitysalustaa. Projektin kannalta onneksaasti ESP32-kehitysalustoja on tilattu toiseen opiskelijaprojektiin kuusi kappaletta. Samasta saapuneesta paketista saadaan kaksi kehitysalustaa testiin KAMK-anturiverkkoa varten.

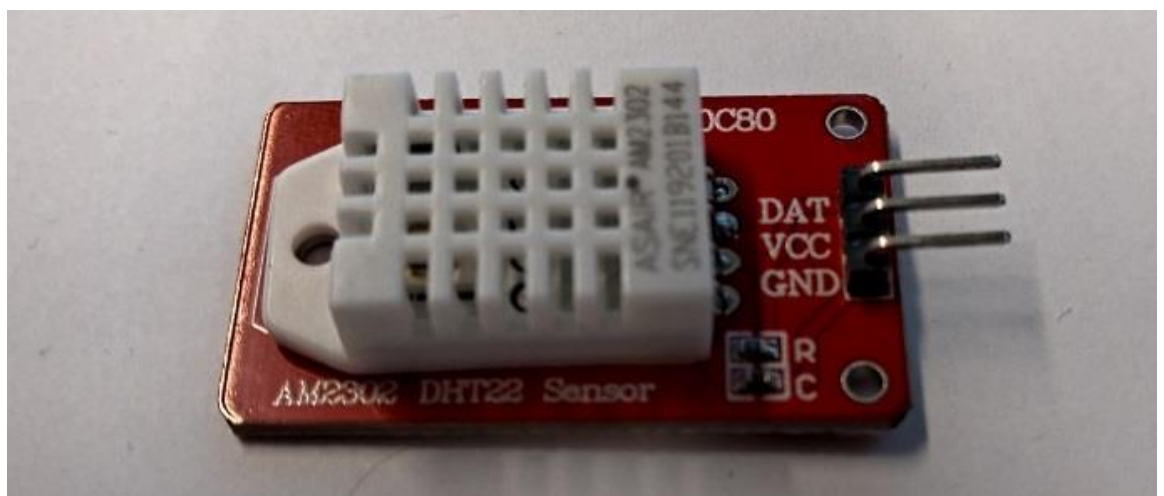
Lyhyen tutustumisen jälkeen todetaan, että ESP32-kehitysalustan ohjelmoiminen Arduino Softwarella (IDE) onnistuu ongelmitta, ja tietokantakommunikointia varten löytyy toimiva ohjelma-kirjasto. Päätetään, että prototyyppiratkaisu tehdään ESP32-kehitysalustaa apuna käyttäen.

## 8.2 Sensoreiden valinta

Käytettävissä olevan kehitysalustan 3,3 V ulostulojännite luo haasteen sensoreiden valinnalle. Matala käyttöjännite karsii käytettävissä olevien sensoreiden valikoimaa, mutta sensoriyksikön kehittämisen välitön aloittaminen katsotaan projektin kannalta tärkeäksi. Näin testaamiseen käytettävissä olevaa aikaa saadaan huomattavasti lisättyä. Sensoreiden valinnassa noudatetaan aloituspäätöksessä asetettua minimivaihtoehtoa sensoreille: lämpötila, ilmankosteus, liike, hiilidioksidi ja valaistus. Lisäksi painoarvoa saa edullinen hinta, käytön suoraviivaisuus ja komponenttien saatavuus. Riittäväksi mittausväliksi katsotaan minuutti, joten sensoreiden nopeuden suhteen vaatimuksia ei juurikaan ole.

### 8.2.1 Lämpötila ja ilmankosteus

Lämpötilan ja ilmankosteuden mittaaminen päätetään toteuttaa DHT22-sensorilla (kuva 10). Datatiedon mukaan käyttöjännite on minimissään 3,3 V, herkkyys lämpötilaa mitattaessa 0,1 astetta celsiusta, herkkyys ilmankosteutta mitattaessa 0,1 %, mittauksen nopeus 2 sekuntia ja kommunikointi tapahtuu digitaalisesti yhden liitäntäpinnin kautta [47]. Sensorille on saatavilla valmiit ohjelmakirjastot Arduino Software (IDE) kirjastonhallinnasta. Sensorin valintaa puoltaa myös halpa hinta sekä hyvä saatavuus. Kirjoitushetkellä Banggood-verkkokauppa listaa sensorin hinnaksi Kiinasta lähetettynä 3,19 euroa. DHT22-sensoreita on myös valmiiksi saatavilla KAMK:n laboratorioista.



Kuva 10. DHT22-sensori

DHT22:n yleisesti tiedossa oleva ongelma on se, että satunnaisesti sensori saattaa lähettää mitauksen arvoksi nollan. Sisätilan ilmaa mitatessa ongelma on kumminkin helposti ohjelmallisesti ohitettavissa, koska sisäilman lämpötila ei normaaleissa olosuhteissa laske nollaan asteeseen. Tällöin satunnaiset nolla-arvot voidaan jättää huomiotta.

### 8.2.2 Liike

Liikkeen tunnistuksen sisällyttäminen prototyyppiratkaisuun osoittautuu haasteelliseksi. Matalan käyttäjännitteen takia vaihtoehtoja etenkin infrapunasensoreiden puolella on varsin vähän. Sensoriksi valikoituu AM312 PIR-sensori (kuva 11). Datalehden mukaan sen käyttäjännitteeksi käy 3,3 V [48]. Kommunikointi kehitysalustalle tapahtuu yhden digitaalisen liitäntäpinnin kautta [48]. AM312:n valintaa puoltaa myös sen halpa hinta. Kirjotushetkellä Amazon-verkkokauppa myy kahden kappaleen pakettia sensorista hintaan 5 euroa.

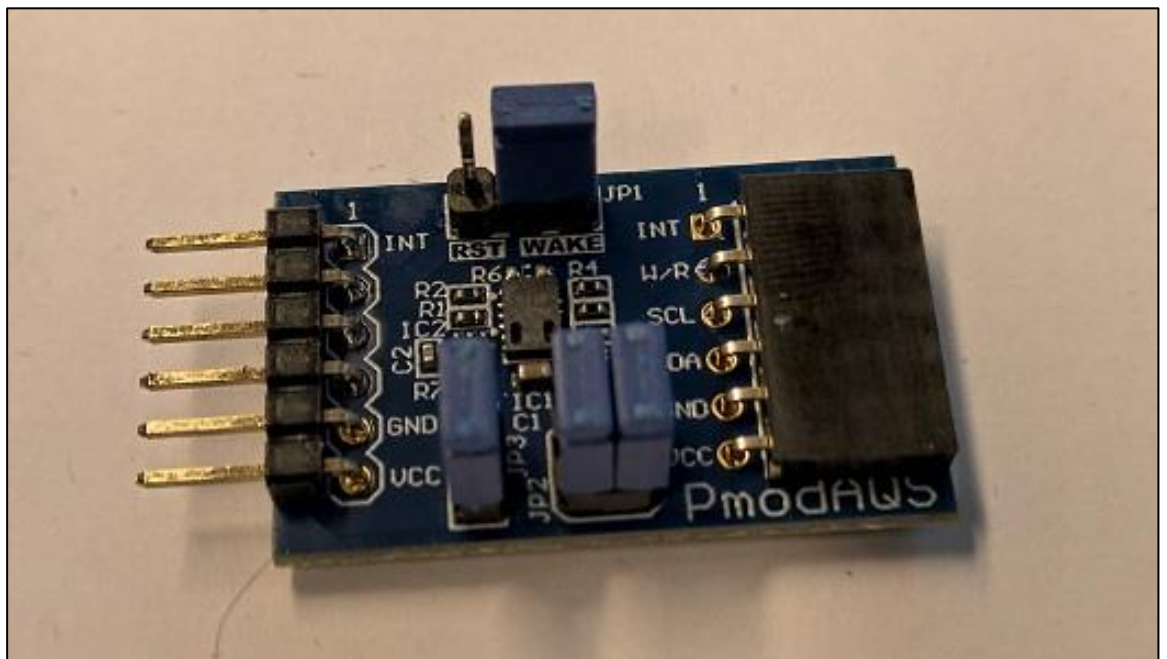


Kuva 11. AM312 PIR-sensori

Infrapunasensoreiden huonona puolena voi pitää lyhyttä liikkeentunnistuksen matkaa. Sensori toimii luotettavasti muutamien metrien etäisyydellä, mutta koko luokkatilan liikkeen tunnistaminen luotettavasti yhdellä PIR-sensorilla on hyvin hankalaa, tai jopa mahdotonta. Prototyyppiratkaisuun PIR-sensori on kumminkin hyvä valinta, koska se toimii hyvänä esimerkkinä liikkeentunnistuksen toiminnallisuudesta.

### 8.2.3 Hiilidioksidi

Hiilidioksiditasoja mittaamaan valitaan Pmod AQS digitaalinen sisäilmanlaadun sensori (kuva 12). Se pohjautuu 3,3 V käyttöjännitteellä toimivaan CCS811-sensoriin ja samalla piirilevyllä sijaitsevaan mikrokontrolleriin, joka mahdollistaa kommunikoinnin I2C-väylän kautta [49]. Sensori ei suoraan havaitse hiilidioksidin määrää ilmassa, vaan se mittaa lukuisia orgaanisia yhdisteitä ja prosessoi omalla mikrokontrollerillaan niistä vastaavan CO<sub>2</sub>-arvon [49]. Kirjoitushetkellä CCS811-sensorin hinta Banggood-verkkokaupassa on 10 euroa. Ohjelmakirjasto CCS811-sensoria varten on saatavilla suoraan Arduino Softwaren (IDE) kirjastonhallinnasta.

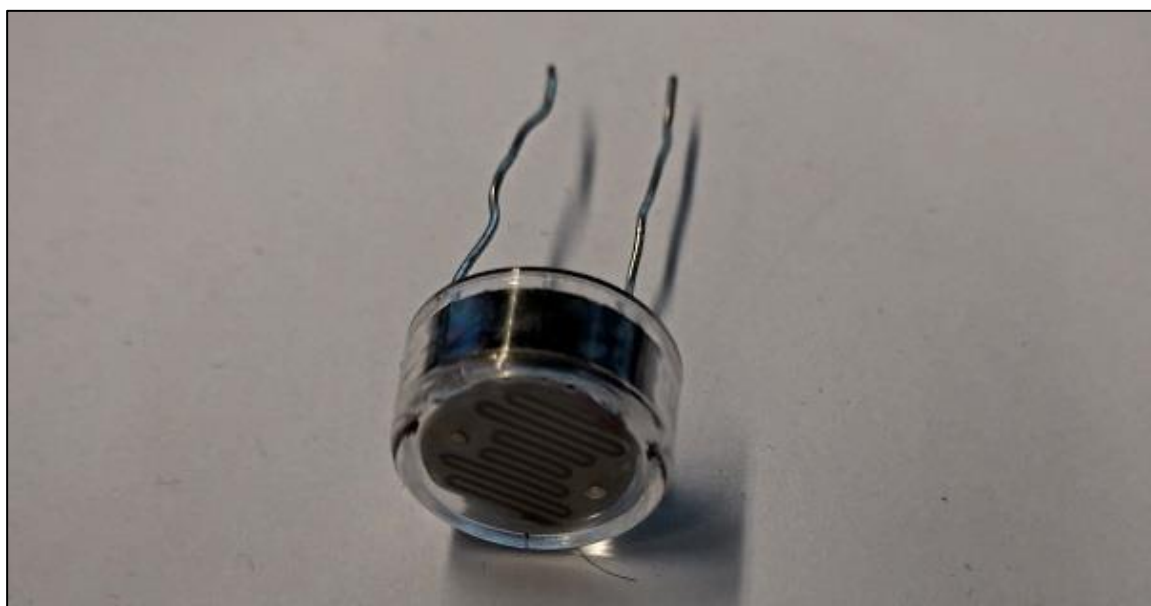


Kuva 12. PmodAQS sisäilmanlaadun sensori

CCS811-sensori on matalan käyttöjännitteensä takia hyvä valinta myös paristokäyttöisiin sensoriyksiköihin. Monesta muusta CO<sub>2</sub>-sensorista poiketen sen käyttöönotto on aikaa vievä prosessi. Sensorin vaatii 48 tunnin "burn-in" -vaiheen mahdollisimman puhtaassa ilmassa toimiakseen oikein [49]. Lisäksi sensoria ei ole suunniteltu toimimaan vapaassa ilmassa, vaan se tarvitsee oikeanlaisen koteloinnin [50].

#### 8.2.4 Valoisuus

Valoisuuden mittauksessa pääasiallinen painopiste on havaita, onko monitoroitavassa tilassa valot päällä vai ei. Sensoriksi valikoituu NORPS-12-valovastus (LDR) (kuva 13). Datalehden mukaan sen vastus muuttuu näkyvän valon mukaan [51]. Valovastuksen mittaustarkkuus riippuu siis siihen liitetyn mikrokontrollerin AD-muuntimen resoluutiosta. Digikey.fi-verkkokauppa listaa kirjoitushetkellä NORPS-12-vastuksen hinnaksi yksittäin ostettuna 3 euroa.

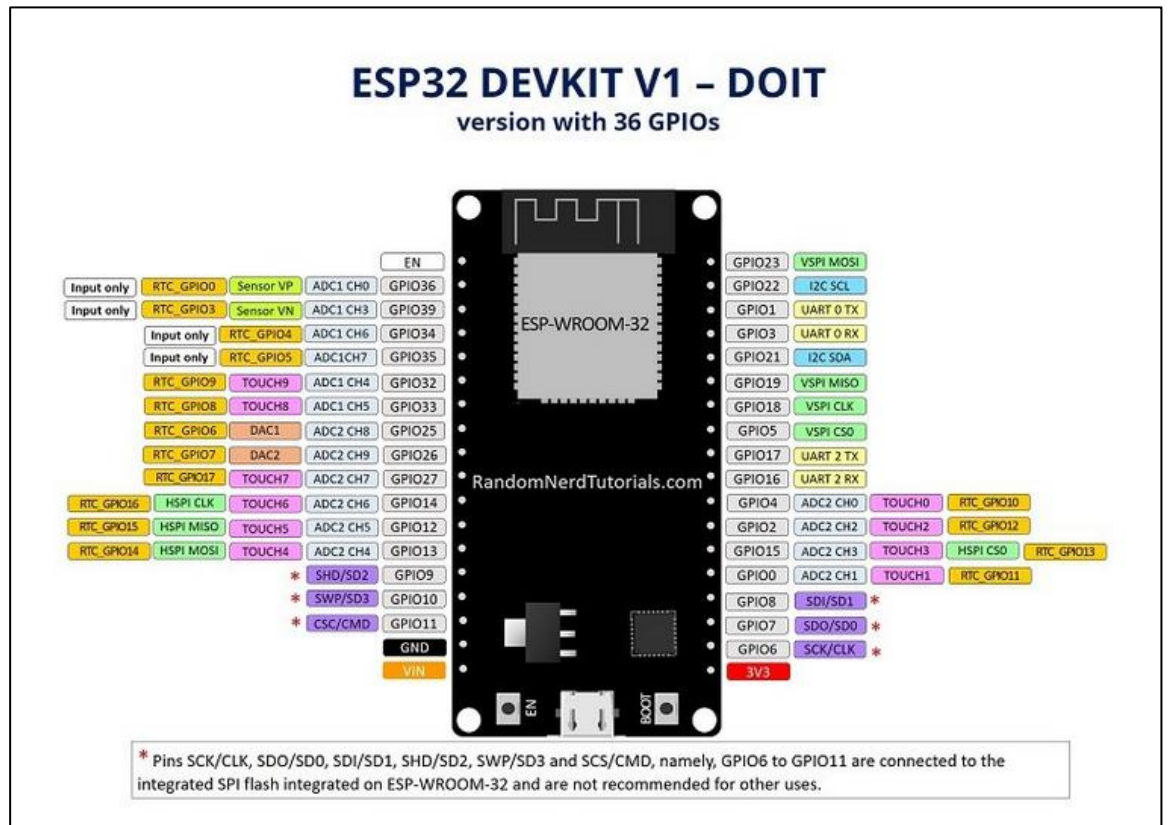


Kuva 13. NORPS-12 valovastus

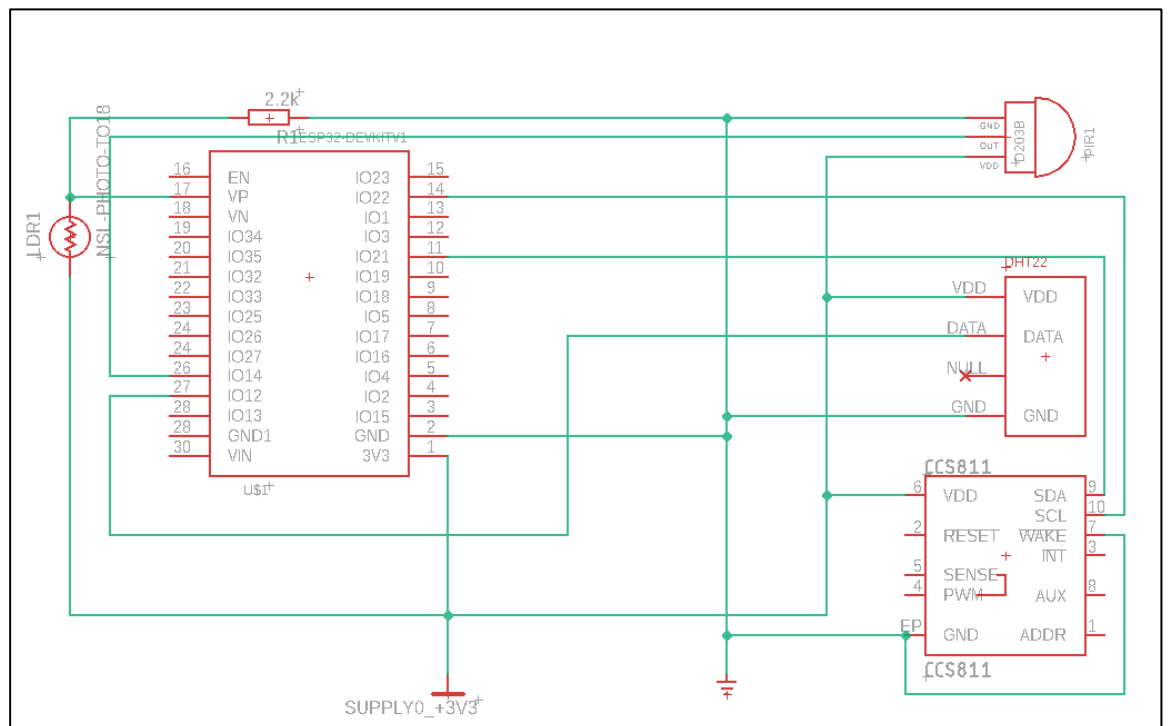
#### 8.3 Sensoriyksikön kytkennät

Sensoriyksikön prototyypin kytkentä päätetään tehdä kytkentäalustalle. Ennen varsinaisen kytkennän suunnittelua etsitään käytössä olevan kehitysalustan (ESP32 DEVKIT V1) kytkentäkaavio (kuva 14).

Kytkenäkaaviosta etsitään vaatimukset täyttävä liitäntäpinni jokaiselle liitettävälle sensorille. Valoisuuden mittauksessa käytettävää LDR-vastusta varten valitaan GPIO-pinni, jonka takaa löytyy A/D-muunnin (analogia-digitaalimuunnin). Kytkentä tehdään kytkentäalustalle, ja siitä piirretään kytkentäkaavio Autodesk Eagle -ohjelmaa apuna käyttäen (kuva 15).



Kuva 14. ESP32 DEVKIT V1 kytkentäkaavio [52].



Kuva 15. Sensoriyksikön kytkentäkaavio

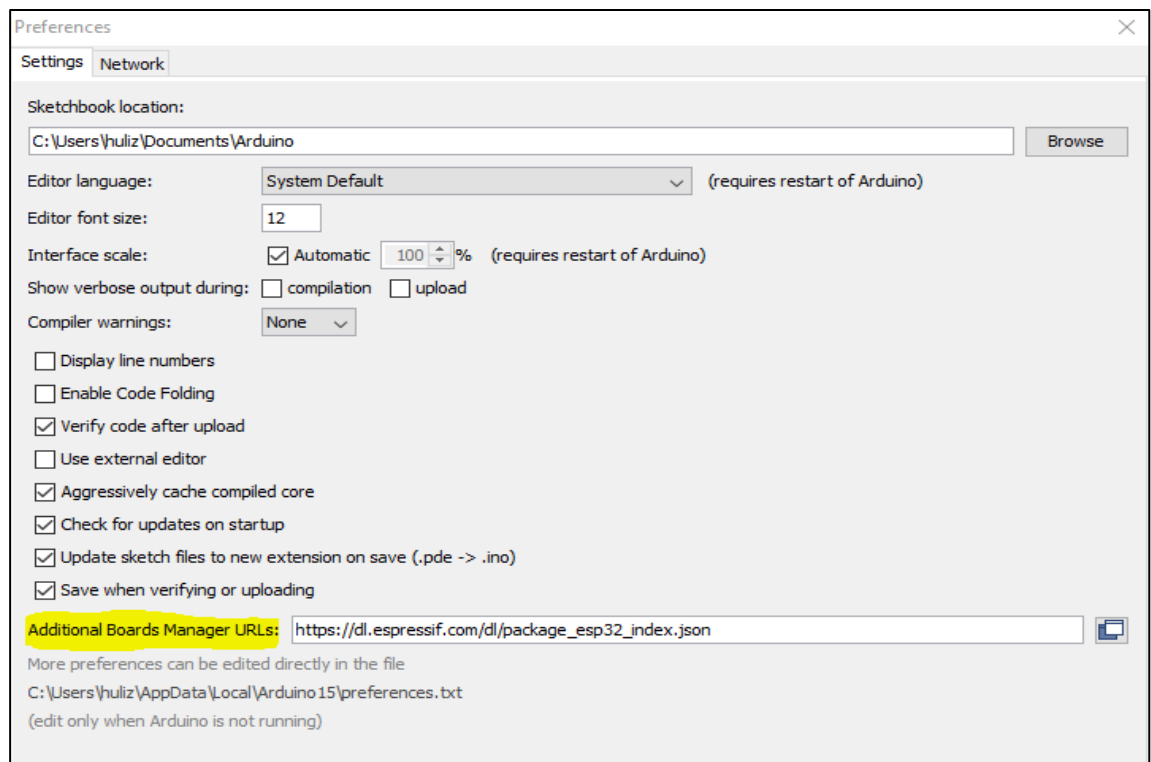
LDR-vastuksen resistanssin määrittämiseksi kytkentään lisätään 2,2 k $\Omega$  alavetovastus. Vastuksen avulla on helppo laskea LDR-vastuksen resistanssi Kirchhoffin jännitelakia apuna käyttäen.

#### 8.4 Sensoriyksikön ohjelmointi

Ennen ohjelmakoodin kirjoittamista tarvitaan Arduino Softwareen (IDE) tuki kehitysalustan ohjelmointia varten. Ohjelmointialustan lisäksi on ladattava tarvittavat kirjastot sensoreille sekä kommunikoinnille InfluxDB:n kanssa.

##### 8.4.1 Kehitysalustan lisääminen Arduino Softwareen (IDE)

Kehitysalustan lisääminen Arduino Softwareen (IDE) onnistuu valitsemalla ”File” -> ”Preferences”, ja lisäämällä halutun kehitysalustan linkki kohtaan ”Additional Boards Manager URLs:” (kuva 16). Jos samaan tekstikenttään on lisätty aiemmin jonkin muun kehitysalustan linkki, niin uuden linkin lisääminen onnistuu erottelemalla linkit pilkulla.



Kuva 16. Arduino Software (IDE) ja ESP32 kehitysalusta



Asetusten muokkaamisen jälkeen valitaan alavetovalikosta "Tools" -> "Board" -> "Boards Manager". Hakukenttään kirjoitetaan esp32, ja listasta valitaan asennettavaksi Espressif Systemsin paketti ESP32-kehitysalustalle (kuva 17). Paketin asennuksen jälkeen valitaan "Board"-valikosta kehitysalustaksi "ESP32 Dev Module", ladataan esimerkkikoodi ja testataan, että ohjelmakoodin lataaminen kehitysalustalle onnistuu ongelmitta.



Kuva 17. Boards manager

#### 8.4.2 Tarvittavien ohjelmakirjastojen lisääminen Arduino Softwareen (IDE)

Käytetyistä ohjelmakirjastoista lähes kaikki on saatavilla suoraan Arduino Softwareen (IDE) kirjastonhallinnasta ("Sketch" -> "Include Library" -> "Manage Libraries..."). Se noudattaa samaa kaavaa kuin edellisessä osiossa läpi käyty Boards Manager. Sensoreista ainoastaan Digilent Pmod AQS ei toimi kirjastonhallinnasta löytyvällä versiolla. Toimiva kirjasto ladataan valmistajan omalta internet-sivuilta [53]. Kirjaston lataamisen jälkeen se lisätään Arduino Softwareen (IDE) valitsemalla alavetovalikosta "Sketch" -> "Include Library" -> "Add .ZIP Library..." ja valitsemalla ladattu kirjasto.

InfluxDB-kommunikointia varten ladataan Thomas Bruenin Influx-Arduino-kirjasto. Se mahdollistaa datan kirjoittamisen InfluxDB-aikasarjatietokantaan ESP32-kehitysalustalla. Influx-Arduino mahdollistaa myös salatun liikenteen tietokantaan. [54.]

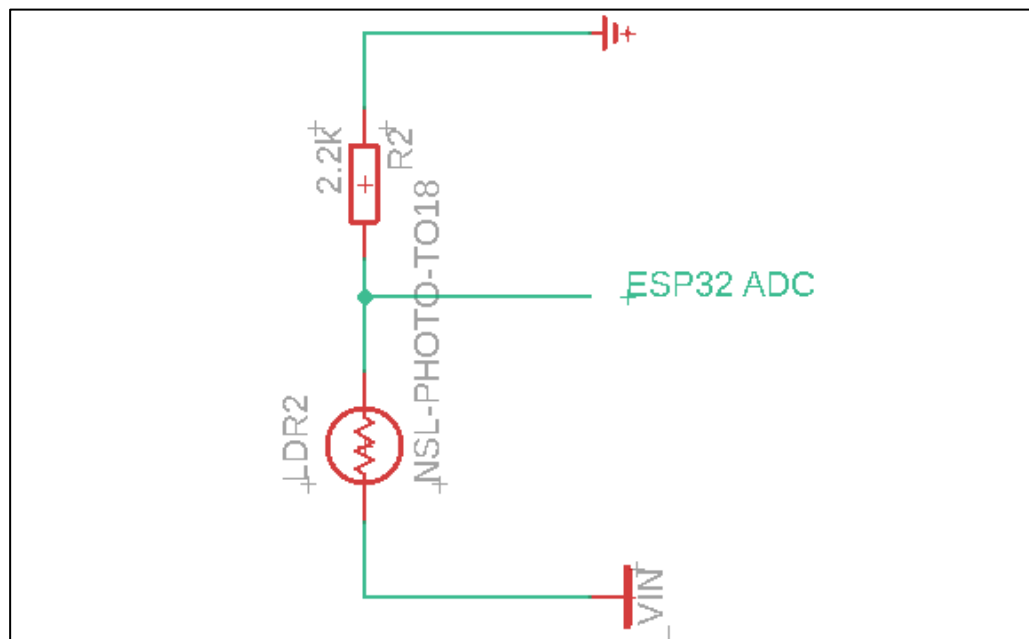
### 8.4.3 Ohjelmakoodi

Ohjelmakoodin (liite 1) pohjaksi valitaan Influx-Arduino-kirjaston mukana tullut WritePoints-esimerkki. Sen avulla testataan, että yhteys ja kirjoitus tietokantaan onnistuu ongelmitta. Tietokantayhteys todetaan toimivaksi, ja seuraavana siirrytään sensoreiden ohjelmakoodiin. Lukuun ottamatta valoisuuden mittausta käytetään kaikkien sensoreiden ohjelmakoodin pohjana ohjelmakirjaston mukana tulleita esimerkkejä. Jokaisen sensorin koodi testataan erikseen ja lopuksi kaikki koostetaan samaan koodiin.

Valoisuuden mittauksessa käytetään pohjana instructables.com sivuston vastaavaa Arduinolle tehtyä projektia [55]. Luksien määrittämiseksi on mitattava LDR-vastuksen resistanssi ESP32 AD-muunninta apuna käyttäen. 12-bittinen AD-muunnin antaa pinnin arvoksi luvun 0–4095. Se voidaan muuttaa jännitteeksi kaavalla 1:

$$V_{out} = analogread \times \frac{3.3V}{4096} = \frac{analogread}{\frac{4096}{3.3V}} = \frac{analogread}{1241} \quad (1)$$

Kun AD-muuntimelle menevä jännite on tiedossa, jää kytkennästä ratkaistavaksi LDR-vastuksen resistanssi (kuva 18).



Kuva 18. LDR-kytkentä

Resistanssi saadaan laskettua Kirchhoffin jännitelain avulla (2):

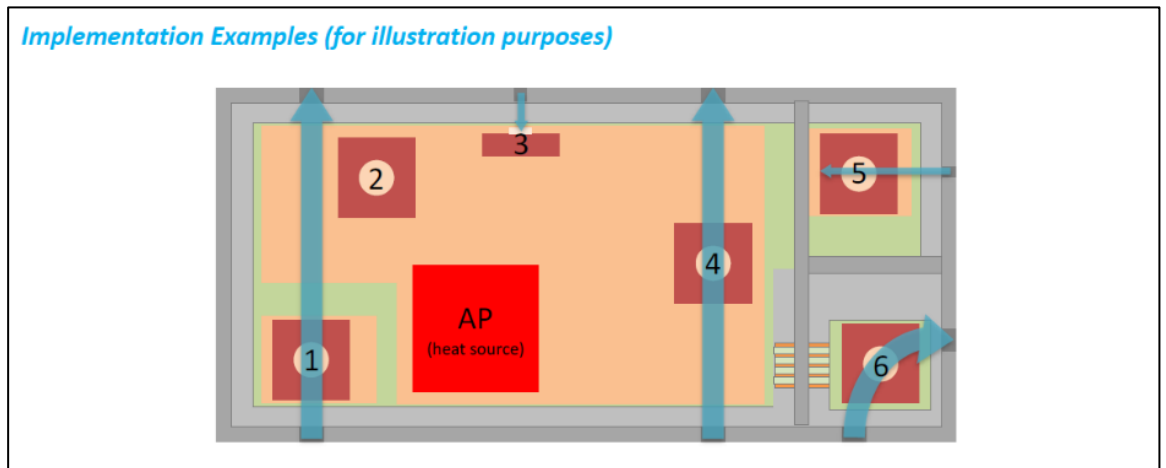
$$V_{OUT} = \frac{R_2}{R_{LDR2} + R_2} \times V_{IN} \quad (2)$$

$$R_{LDR} = \frac{R_2 \times V_{IN} - R_2 \times V_{OUT}}{V_{OUT}}$$

Instructables.com-sivuston projektissa vastuksen arvoja on vertailtu luksimittarin antamiin arvoihin, ja niiden perusteella on laskettu yhtälö luksiarvojen saamiseksi [55]. Valmis yhtälö lisätään ohjelmakoodiin, ja ohjelman ilmoittamia luksiarvoja vertaillaan PeakTech 5025 -mittarin ilmoittamiin arvoihin. Havaitaan, että arvot ovat varsin hyvin linjassa luksimittarin kanssa. Tässä vaiheessa päätetään, että käytetään valmista yhtälöä, koska luokkatilan valaistusta havainnoidessa ei absoluuttiselle arvolle ole tarvetta.

## 8.5 Sensoriyksikön kotelointi

Alkuperäisessä suunnitelmassa koteloinnille ei annettu paljoa painoarvoa. Koodin valmistuttua havaitaan, että kytkentäalustalle rakennetun prototyyppiratkaisun ilmoittamat hiilidioksidiarvot hyppivät liian suuriin lukemiin (jopa 2500 ppm). Asiaa tutkitaan, kunnes huomataan, että CCS811-sensorin valmistajan sivuilta löytyy kotelointia koskeva dokumentaatio. Dokumentaation mukaan parhaan toiminnan takaamiseksi sensorin pitäisi olla kotelossa, jossa huoneilmassa olevat partikkelit eivät pääse suoraan laskeutumaan sensorin päälle [50]. Lisäksi sensorin pitäisi olla mahdollisimman hyvin eristetty muista lämpöä tuottavista komponenteista [50]. Kotelointiohjeen mukaan numerolla 6 merkitty sijoitus olisi sensorin toiminnan kannalta paras vaihtoehto (kuva 19) [50].



Kuva 19. CCS811 kotelointiohjeet [50].

Koteloinnin tekemisessä avuksi lupautuu konetekniikan insinööriopiskelija Pekka Turunen. Komponenteista otetaan mitat, niiden pohjalta kotelosta tehdään 3d-malli ja se tulostetaan 3d-tulostinta apuna käyttäen. Sensoriyksikön prototyyppi (kuva 20 ja 21) rakennetaan koteloon ja huomataan, että CCS811-sensorin ilmoittamat hiilidioksidiarvot eivät tee enää jyrkkiä hyppyjä.



Kuva 20. Sensoriyksikön kotelo, jonka sisälle on asetettu kytkentäalusta



Kuva 21. Sensoriyksikön prototyyppi toiminnassa opettajan pöydällä.

#### 8.6 Sensoriyksikön kokonaiskustannukset

Sensoriyksikön kokonaiskustannukset saadaan pidettyä varsin maltillisina (taulukko 11). Kokonaiskustannuksessa käytetyt hinnat ovat yksittäin ostettujen komponenttien hintoja. Suurempia eriä ostaessa komponenttien hankintahinta laskee hieman.

Nimike	Kauppa	Hinta
ESP32 dev board	Banggood.com	10 €
Microusb virtalähde	Banggood.com	5 €
DHT22	Banggood.com	3,20 €
AM312	Amazon.com	2,50 €
CCS811	Banggood.com	10 €
NORPS-12	Digikey.fi	3 €
Kotelo (3D-tulostettu)		2 €
<b>Yhteensä</b>		<b>36 €</b>

Taulukko 11. Sensoriyksikön kokonaiskustannukset

## 8.7 Havaitut ongelmat

Infrastruktuurin ja sensoriyksikön prototyypin rakennus sujui lähestulkoon ongelmitta, mutta liikkeen tunnistuksen osalta kaksi ongelmaa jää selvittämättä.

Ensimmäinen ongelmista liittyy käytössä olevaan AM312 PIR -sensoriin. Se tunnistaa välillä liikettä, vaikka sitä ei olisi. Ongelmaa yritetään korjata lisäämällä kytkentään kondensaattori, mutta sillä ei ole vaikutusta haamutunnistuksiin. Suurimmalla osalla testatuista AM312 PIR -sensoreista haamutunnistuksia tapahtuu kumminkin vain muutamia kertoja vuorokaudessa. Ongelmaan ei paneuduta tarkemmin, koska kehitysalustaa on saatavilla myös 5 voltin käyttöjännitettä tarjoavalla piirilevyllä. Tällöin PIR-sensori voidaan korvata jollakin varmatoimisemmalla mallilla.

Toinen työssä havaittu ongelma jää selvittämättä sen satunnaisuuden takia. Sensoriyksiköiden huomataan kaatuvan satunnaisesti. Vikaa yritetään paikallistaa ensin kytkemällä sensoriyksikkö tietokoneeseen ja lukemalla sarjaliikenneportista virheilmoituksia. Kaatuilu kumminkin loppuu, kun sensoriyksikkö on yhdistettynä tietokoneeseen. Lopulta sensoriyksikkö asetetaan luokkatiilaan, jossa kaatuilua tapahtuu eniten. Se yhdistetään RasPiin, jonka avulla debug-ulostulo tallennetaan tekstitiedostoon. Virheviesteistä havaitaan, että kaatuilu on liitännäistä WLAN-yhteyteen ja PIR-sensorin aiheuttamaan keskeytykseen. Kaatuilua tapahtuu kumminkin niin harvoin, että ratkaisua ongelmaan ei ehditä löytää aikataulun puitteissa. Parhaimmillaan sensoriyksikkö toimii ongelmitta jopa 3–4 viikkoa, mutta välillä kaatuilua tapahtuu jopa useamman kerran päivässä. PIR-sensorin keskeytyksen poistaminen ohjelmakoodista lopettaa sensoriyksikön satunnaisen kaatumisen, joten siitä päätellen ongelma on jotenkin keskeytykseen liitännäinen.

## 9 KAMK-anturiverkko: tulevaisuus

KAMK-anturiverkon infrastruktuuri saadaan rakennettua suunnitelman mukaisesti valmiiksi. Toimivan infrastruktuurin päälle on hyvä lähteä kehittämään lopullista versiota sensoriyksiköstä. Sensoriyksikön prototyyppi kehitetään varsin tuoreelle kehitysalustalle, mikä voi olla osasyynä sille, että kaikkia havaittuja ongelmia ei saada ratkaistua. Kehitys IoT-kehitysalustojen osalta on kumminkin nopeaa ja uudet firmware-päivitykset voivat itsessään tuoda ratkaisun ongelmiin. Ennen kuin KAMK-anturiverkkoa päästään hyödyntämään luokkatilojen käyttöasteen seurantaan, tarvitaan valmis tuotantoon kelpaava sensoriyksikkö, jonka voi viedä kampuksen jokaiseen monitoroitavaan luokka- tai laboratoriotilaan.

Sensoriyksikön loppukehitys tarjoaa hyvän kohteen tuleville opiskelijaprojekteille. Kun sensoriyksikköä vaivaavat pienet ongelmat saadaan ratkaistua, on vuorossa piirilevyn suunnittelu, joka palvelee tieto- ja viestintätekniikan koulutusohjelman oppimistavoitteita. Piirilevysuunnittelun kanssa käsi kädessä kulkeva koteloinnin suunnittelu taas tuo hyvän kohteen poikkialaiselle projektitoiminnalle esimerkiksi konetekniikan koulutusohjelman kanssa. Valmiin sensoriyksikön koostaminen komponenteista valmiille piirilevypohjalle on oiva työ opintoaan aloittaville tekniikan alan opiskelijoille. Lisäksi sensoriyksikön prototyyppit ja kehitysversiot tuottavat analysoitavaa dataa tietokantaan. Helposti opetuksen käyttöön saatavilla oleva tosielämän data mahdollistaa Datasta tekoälyyn -koulutusohjelmalle hyvän harjoitusalueen datan analysointiharjoituksille.

Ennen lopullisen sensoriyksikön kehittämistä olisi syytä ottaa huomioon KAMK-anturiverkon tuotaman datan käyttäminen taloautomaation ohjaamisessa. Valittu kehitysalusta tarjoaa hyvät laajentamismahdollisuudet sensoreiden suhteen. Esimerkiksi ilmanpainesensorin avulla voitaisiin seurata ilmanpaine-eroja rakennuksen eri huonetilojen välillä. Olisikin varsin ideaalinen tilanne, jos pääosin opiskelijavoimin rakennettu sensoriverkko voisi palvella useamman koulutusohjelman opetustarpeita, olla apuna luokkatilojen käyttöasteen seurannassa ja tehostaa taloautomaation toimintaa.

## 10 Yhteenveto

Työn tavoitteena oli luoda Kajaanin Ammattikorkeakoulu Oy:n tiloihin prototyyppiratkaisu luokkatilojen käyttöasteen seurannan mahdollistavasta sensoriverkosta. Tehtävä jakautui kahteen suurempaan osa-alueeseen: infrastruktuurin rakentaminen ja sensoryksikön prototyypin rakentaminen.

IoT-laitteiden nopea kehitys ja halpa hinta takaavat sen, että sensoriverkkoja rakennetaan tulevaisuudessa entistä enemmän. KAMK-anturiverkko tarjosi loistavan mahdollisuuden päästä rakentamaan sensoriverkko kokonaisuutena. Aihe sopi erinomaisesti tieto- ja viestintätekniikan insinööriopiskelijan opinnäytetyön aiheeksi. Infrastruktuurin rakentamisessa saatiin opetuksen kautta saatujen taitojen lisäksi hyödyntää myös omaa harrastuneisuutta verkko- ja palvelinteknologiaista.

Itse työn suoritus onnistui hyvin. Infrastruktuuri saatiin rakennettua odotettua sujuvammin, mistä voi osittain kiittää KamIT Tietohallinnon kanssa yhteistyössä tehtyä VPN-yhteyttä. Sensoryksikön prototyypin osalta pari havaittua ongelmaa jäi selvittämättä. Työn laajuuden ja tiukan aikataulun puitteissa sensoryksikön prototyyppi kumminkin täyttää sille asetetut vaatimukset. Dokumentoinnissa suurimmat haasteet kohdattiin teoriaosuudessa. Työ haluttiin suorittaa aikataulussa, joten teoriaosuuteen sisällytettäviä aiheita piti valikoida tarkkaan.

Lopuksi haluan kiittää työn toimeksiantajaa Kajaanin Ammattikorkeakoulu Oy:tä sekä kaikkia projektissa apuna olleita henkilöitä. Toivomuksena on, että KAMK-anturiverkko otetaan osaksi opetusta, jolloin sensoryksikön kehitys sekä sensoriverkon laajentaminen voidaan viedä loppuun opiskelijavoimin.



## Lähteet

1. Kajaanin ammattikorkeakoulun supertietokone on käynnistynyt [Internet]. 2018 [Viitattu 27.01.2020]. (Kajaanin kaupunki). Saatavilla: <http://www.kajaani.fi/fi/tiedote/kajaanin-ammattikorkeakoulun-supertietokone-on-kaynnistynyt>
2. Kajaanin ammattikorkeakoulu joutuu yt-neuvotteluihin miljoonatappion jälkeen [Internet]. [Viitattu 27.01.2020]. (Yle Uutiset). Saatavilla: <https://yle.fi/uutiset/3-10587910>
3. Foote KD. A Brief History of the Internet of Things [Internet]. 2016 [Viitattu 28.01.2020]. (DATAVERSITY). Saatavilla: <https://www.dataversity.net/brief-history-internet-things/>
4. Hinden RM, Deering SE. IP Version 6 Addressing Architecture [Internet]. [Viitattu 28.01.2020]. Saatavilla: <https://tools.ietf.org/html/rfc4291>
5. We bring you the 10 most popular prototyping and development boards in 2020 [Internet]. [Viitattu 28.01.2020]. Saatavilla: <https://www.youngwonks.com/blog/Top-10-IoT-boards-for-2019>
6. ammattiopisto K seudun. Topologiat. 2010 [Internet]. [Viitattu 28.01.2020]. Saatavilla: <http://www.koudata.fi/node/585>
7. History | Wi-Fi Alliance [Internet]. [Viitattu 28.01.2020]. (<https://www.wi-fi.org/who-we-are/history>). Saatavilla: <https://www.wi-fi.org/who-we-are/history>
8. Lee J-S, Su Y-W, Shen C-C. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In Ieee; 2007. s. 46–51.
9. 802.11ac vs 802.11n - What's the difference between the Wi-Fi standards? [Internet]. 2013 [Viitattu 03.02.2020]. (Trusted Reviews). Saatavilla: <https://www.trusted-reviews.com/opinion/802-11ac-vs-802-11n-what-s-the-difference-2905251>
10. OFFICIAL IEEE 802.11 WORKING GROUP PROJECT TIMELINES - 2020-01-26 [Internet]. 2020. [Viitattu 03.02.2020]. Saatavilla: [http://grouper.ieee.org/groups/802/11/Reports/802.11\\_Timelines.htm](http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm)

11. journals I, Osman RBMA 2Amin Babiker A. Mustafa, 3Ashraf A. A Comparison between IEEE 802.11 n and ac Standards [Internet]. Figshare; 2015. [Viitattu 03.02.2020]. Saatavilla: <https://search.datacite.org/works/10.6084/M9.FIGSHARE.1564603>
12. Our History [Internet]. [Viitattu 03.02.2020]. (Bluetooth Technology Website). Saatavilla: <https://www.bluetooth.com/about-us/our-history/>
13. Sponås JG. Things You Should Know About Bluetooth Range [Internet]. [Viitattu 03.02.2020]. Saatavilla: <https://blog.nordicsemi.com/getconnected/things-you-should-know-about-bluetooth-range>
14. Al-Kashoash HA, Kemp AH. Comparison of 6LoWPAN and LPWAN for the Internet of Things. 2016;13(4):268–74.
15. Ergen SC. ZigBee/IEEE 802.15. 4 Summary. 2004;10:17.
16. NB-IoT | IoT-ratkaisut | Telia yrityksille [Internet]. [Viitattu 10.02.2020]. Saatavilla: <https://www.telia.fi/yrityksille/iot/yhteydet/nb-iot>
17. 2012 © 3GPP. 3GPP Standards for the Internet-of-Things.
18. Semtech Acquires Wireless Long Range IP Provider Cycleo [Internet]. [Viitattu 10.02.2020]. (Design And Reuse). Saatavilla: <https://www.design-reuse.com/news/28706/semtech-cycleo-acquisition.html>
19. About LoRa Alliance [Internet]. [Viitattu 10.02.2020]. (<https://lora-alliance.org/about-lora-alliance>). Saatavilla: <https://lora-alliance.org/about-lora-alliance>
20. IoT-verkkoteknologioiden vertailu [Internet]. [Viitattu 10.02.2020] Metropolia Ammattikorkeakoulu; 2017. Saatavilla: <http://urn.fi/URN:NBN:fi:amk-201705178730>
21. Ojanperä V. Nyt IoT-dataa ilman kuukausi- tai vuosimaksuja [Internet]. [Viitattu 10.02.2020]. Saatavilla: <https://etn.fi/index.php/13-news/6383-nyt-iot-dataa-ilman-kuukausi-tai-vuosimaksuja>
22. Femtoduino: an ultrasmall (20.7x15.2 mm) libre Arduino compatible board [Internet]. [Viitattu 11.02.2020]. Saatavilla: <http://www.varesano.net/projects/hardware/Femtoduino>

23. Azzola F. 10 Arduino IDE alternative to start programming [Internet]. 2018 [Viitattu 11.02.2020]. (Surviving with Android). Saatavilla: <https://www.survivingwithandroid.com/10-arduino-ide-alternative-to-start-programming/>
24. Arduino Software (IDE) [Internet]. [Viitattu 11.02.2020]. Saatavilla: <https://www.arduino.cc/en/Guide/Environment>
25. History and background of the Raspberry Pi - Raspberry Pi Computer Architecture Essentials [Internet]. 2016 [Viitattu 11.02.2020]. Saatavilla: [https://subscription.packtpub.com/book/hardware\\_and\\_creative/9781784397975/1/ch01lvl1sec08/history-and-background-of-the-raspberry-pi](https://subscription.packtpub.com/book/hardware_and_creative/9781784397975/1/ch01lvl1sec08/history-and-background-of-the-raspberry-pi)
26. RaspberryPI models comparison [Internet]. [Viitattu 11.02.2020]. Saatavilla: <http://socialcompare.com/en/comparison/raspberrypi-models-comparison>
27. Setting up your Raspberry Pi [Internet]. [Viitattu 11.02.2020]. Saatavilla: <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/2>
28. 1. The Arduino Family - Arduino: A Technical Reference [Book] [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://www.oreilly.com/library/view/arduino-a-technical/9781491934319/ch01.html>
29. Arduino Products [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://www.arduino.cc/en/Main/Products>
30. Arduino MKR WIFI 1010 [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://store.arduino.cc/arduino-mkr-wifi-1010>
31. Arduino Nano 33 IoT [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://store.arduino.cc/arduino-nano-33-iot>
32. ARDUINO UNO WiFi REV2 [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://store.arduino.cc/arduino-uno-wifi-rev2>
33. Espressif milestones [Internet]. [Viitattu 17.02.2020]. Saatavilla: <https://www.espressif.com/en/company/about-us/milestones>

34. New Chip Alert: The ESP8266 WiFi Module (It's \$5) [Internet]. 2014 [Viitattu 18.02.2020]. (Hackaday). Saatavilla: <https://hackaday.com/2014/08/26/new-chip-alert-the-esp8266-wifi-module-its-5/>
35. A Comparison of the New ESP32-S2 to the ESP32 [Internet]. [Viitattu 02.03.2020]. Saatavilla: <https://maker.pro/esp8266/tutorial/a-comparison-of-the-new-esp32-s2-to-the-esp32>
36. Time Series DBMS [Internet]. [Viitattu 02.03.2020]. Saatavilla: <https://db-engines.com/en/article/Time+Series+DBMS>
37. Method of calculating the scores of the DB-Engines Ranking [Internet]. [Viitattu 02.03.2020]. Saatavilla: [https://db-engines.com/en/ranking\\_definition](https://db-engines.com/en/ranking_definition)
38. InfluxDB System Properties [Internet]. [Viitattu 02.03.2020]. Saatavilla: <https://db-engines.com/en/system/InfluxDB>
39. What is virtualization? [Internet]. [Viitattu 09.03.2020]. (Opensource.com). Saatavilla: <https://opensource.com/resources/virtualization>
40. Installing InfluxDB OSS | InfluxData Documentation [Internet]. [Viitattu 09.03.2020]. Saatavilla: <https://docs.influxdata.com/>
41. Grafana Requirements [Internet]. [Viitattu 09.03.2020]. (Grafana Labs). Saatavilla: <https://grafana.com/docs/grafana/latest/installation/requirements/>
42. Hardware sizing guidelines | InfluxData Documentation [Internet]. [Viitattu 09.03.2020]. Saatavilla: <https://docs.influxdata.com/>
43. Grafana documentation [Internet]. [Viitattu 16.03.2020]. (Grafana Labs). Saatavilla: <https://grafana.com/docs/grafana/latest/>
44. InfluxDB 1.7 documentation | InfluxData Documentation [Internet]. [Viitattu 16.03.2020]. Saatavilla: <https://docs.influxdata.com/>
45. Joker.com FAQ - Let's Encrypt Support [Internet]. [Viitattu 16.03.2020]. Saatavilla: [https://joker.com/faq/content/6/496/en/let\\_s-encrypt-support.html](https://joker.com/faq/content/6/496/en/let_s-encrypt-support.html)

46. IPsec VPN Overview [Internet]. [Viitattu 16.03.2020]. Saatavilla: [https://www.juniper.net/documentation/en\\_US/junos/topics/topic-map/security-ipsec-vpn-overview.html](https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ipsec-vpn-overview.html)
47. aosong. Digital-output relative humidity & temperature sensor/module.
48. AM312 Datasheet. NANYANG SENBA OPTICAL AND ELECTRONIC CO. LTD.;
49. CCS811. AMS AG;
50. Mechanical Considerations for CCS811. Cambridge CMOS Sensors;
51. WWW.LUNAINC.COM. NORPS-12. Luna Optoelectronics; 2016.
52. ESP32 Pinout Reference: Which GPIO pins should you use? [Internet]. 2019 [Viitattu 24.03.2020]. (Random Nerd Tutorials). Saatavilla: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>
53. Pmod AQS [Reference.Digilentinc] [Internet]. [Viitattu 30.03.2020]. Saatavilla: <https://reference.digilentinc.com/reference/pmod/pmodaqs/start>
54. Influx-Arduino [Internet]. [Viitattu 30.03.2020]. Saatavilla: <https://github.com/teebr/Influx-Arduino>
55. Measuring Light Using Light Sensor [Internet]. [Viitattu 30.03.2020]. (Instructables). Saatavilla: <https://www.instructables.com/id/Measuring-Light-Using-Light-Sensor/>

## Liitteet

```
#include <WiFi.h>
#include "InfluxArduino.hpp"
#include "InfluxCert.hpp"
#include <DHT.h>
#include <DHT_U.h>
#include <math.h>
#include <string.h>
#include "Wire.h"
#include "PmodAQS.h"
#include <esp_attr.h>

#define PIR 14
#define LDR 36
#define DHTPIN 27
#define DHTTYPE DHT22

int motion = 0;
int stopmotion = 0;

DHT dht(DHTPIN, DHTTYPE);

float readDHTTemperature() {
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow
    sensor)
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // If no reading try to read again to make sure it isn't occasional
    zero value from DHT-22
    // Read temperature as Fahrenheit (isFahrenheit = true)
    //float t = dht.readTemperature(true);
    // Check if any reads failed and exit early (to try again).
    if (isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return 0;
    }
    else {
        Serial.print("TEMP: ");
        Serial.println(t);
        return t;
    }
}

float readDHTHumidity() {
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow
    sensor)
    float h = dht.readHumidity();
    // If no reading try to read again to make sure it isn't occasional
    zero value from DHT-22
    if (isnan(h)) {
        Serial.println("Failed to read from DHT sensor!");
        return 0;
    }
    else {
        Serial.print("HUM: ");
        Serial.println(h);
    }
}
```

```

        return h;
    }
}

float readLux() {
    float input = analogRead(LDR);
    float vinput = input/1241.2;
    float rldr = (7210-vinput*2200)/vinput;
    float lux= (pow( rldr, (1/-0.8616)))/(pow( 10, (5.118/-0.8616)));

    if (isnan(lux)) {
        Serial.println("Failed to read from sensor!");
        return 0;
    }
    else {
        Serial.print("LUM: ");
        Serial.println(lux);
        return lux;
    }
}

// Checks if motion was detected
void IRAM_ATTR detectsMovement() {
    Serial.println("MOTION DETECTED!!!");
    //Set motion on
    motion = 1;
    // Set stopmotion counter to 60.
    stopmotion = 60;
}

InfluxArduino influx;
//connection/ database stuff that needs configuring
const char KAMK_ROOM[] = "TESTUNIT";
const char WIFI_NAME[] = "SSID";
const char WIFI_PASS[] = "WIFI_PASSWORD";
const char INFLUX_DATABASE[] = "anturiverkko";
const char INFLUX_IP[] = "0.0.0.0";
const char INFLUX_USER[] = "username"; //username if authorization is
enabled.
const char INFLUX_PASS[] = "password"; //password for if authorization
is enabled.
const char INFLUX_MEASUREMENT[] = "esp32_test"; //measurement name for
the database. (in practice, you can use several, this example just uses
the one)just uses the one)

unsigned long DELAY_TIME_US = 5 * 1000 * 1000; //how frequently to send
data, in microseconds
//unsigned long count = 0; //a variable that we gradually increase in
the loop

float tempdht22 = 0;
float humidity = 0;
float lux = 0;
float co2ppm = 0;
float tvocppb = 0;

int reconnectcounter = 0;

PmodAQS ccs;

void setup()

```

```

{
    Serial.begin(115200);

    Serial.println("PmodAQS test");

    if(!ccs.begin(0x5B)){
        Serial.println("Failed to start sensor! Please check your
wiring.");
        while(1);
    }

    //calibrate temperature sensor
    while(!ccs.available());
    ccs.setEnvironmentalData(20,20);

    dht.begin();

    pinMode(LDR, INPUT);

    pinMode(PIR, INPUT);

    attachInterrupt(digitalPinToInterrupt(PIR), detectsMovement,
CHANGE);

    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("WiFi connected!");
    influx.configure(INFLUX_DATABASE,INFLUX_IP); //third argument (port
number) defaults to 8086
    influx.authorize(INFLUX_USER,INFLUX_PASS); //if you have set the
Influxdb .conf variable auth-enabled to true, uncomment this
    // influx.addCertificate(ROOT_CERT); //uncomment if you have
generated a CA cert and copied it into InfluxCert.hpp
    Serial.print("Using HTTPS: ");
    Serial.println(influx.isSecure()); //will be true if you've added
the InfluxCert.hpp file.
}

void loop()
{
    reconnectcounter = 0;
    unsigned long startTime = micros(); //used for timing when to send
data next.

    //Every 5 second loop we subtract from stopmotion counter
    if(stopmotion > 0)
    {
        stopmotion--;
    }
    //Stopmotion counter is zero and we report no motion
    else
    {
        motion = 0;
    }
    //update our field variables

```



```

float ctemp = readDHTTemperature();
float chumidity = readDHTHumidity();
lux = readLux();

if (ctemp!=0)
{
    tempdht22 = ctemp;
}
if (chumidity!=0)
{
    humidity = chumidity;
}

if(ccs.available()){

    uint8_t envhumidity = humidity;
    double envtemp = tempdht22;
    ccs.setEnvironmentalData(envhumidity,envtemp);

    if(!ccs.readData()){
        Serial.print("CO2: ");
        co2ppm = ccs.getCO2();
        Serial.print(co2ppm);
        Serial.print("ppm\nTVOC: ");
        //Serial.print(ccs.getTVOC());
        tvocppb = ccs.getTVOC();
        Serial.print(tvocppb);
        Serial.println("ppb");
    }
    else{
        Serial.println("ERROR!");
        while(1);
    }
}

//write our variables.
char tags[32];
char fields[82];
sprintf(tags,"name=%s",KAMK_ROOM); //write a tag called KAMK_ROOM

sprintf(fields,"temperature=%0.2f,humidity=%0.2f,lux=%0.1f,movement=%i,
co2ppm=%0.1f,tvocppb=%0.1f",tempdht22,humidity,lux,motion,co2ppm,tvocpp
b); //write three fields: temperature, humidity and lux
bool writeSuccessfull =
influx.write(INFLUX_MEASUREMENT,tags,fields);
if(!writeSuccessfull)
{
    Serial.print("error: ");
    Serial.println(influx.getResponse());
    //Usually disconnected wifi causes problems so here we
reconnect
    WiFi.disconnect(true);
    delay(1000);
    WiFi.mode(WIFI_OFF);
    delay(1000);
    Serial.println("Connecting to WiFi...");
    WiFi.mode(WIFI_STA);
    delay(1000);
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);

```

```
        Serial.print(".");
        reconnectcounter++;
        if(reconnectcounter>60)
        {
            //Seems reconnecting didn't work this time. Hopefully it
works next time.
            return;
        }
    }

    while ((micros() - startTime) < DELAY_TIME_US)
    {
        //wait until it's time for next reading. Consider using a low
power mode if this will be a while.
    }
}
```